

OPTIMIZATION OF NONCONVEX SYSTEMS
AND THE SYNTHESIS OF OPTIMUM
PROCESS FLOWSHEETS

By

GEORGE STEPANOPoulos

A DISSERTATION PRESENTED TO THE GRADUATE
COUNCIL OF THE UNIVERSITY OF FLORIDA
IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE
DEGREE OF DOCTOR OF PHILOSOPHY

UNIVERSITY OF FLORIDA
1974

Dedicated to
The Memory of My Father
Nicholas Stephanopoulos

*Man has three ways of acting wisely:
Firstly, on meditation, this is the
noblest; Secondly, on imitation, this
is the easiest; and Thirdly, on
experience; this is the bitterest.*

Confucius

ACKNOWLEDGMENT

The author wishes to express his gratitude to:

The chairman of his supervisory committee, Dr. Arthur W. Westerberg, Associate Professor of Chemical Engineering, for his firm support, sound advice, and guidance throughout what has been the most stimulating and enjoyable period of his academic life;

The members of his supervisory committee: Dr. F.P. May, Professor of Chemical Engineering, Dr. D.W. Kirmse, Assistant Professor of Chemical Engineering, Dr. T.E. Bullock, Associate Professor of Electrical Engineering and Dr. M.E. Thomas, Professor of Industrial and Systems Engineering;

The faculty and staff of the Department of Chemical Engineering for their assistance;

The National Science Foundation, which provided financial support through the grants GK-18633 and 41606 and the Graduate School for a research assistantship;

Jeanette for her encouragement through the lows and the sharing of the highs of the life.

TABLE OF CONTENTS

	<u>Page</u>
ACKNOWLEDGMENTS	iii
LIST OF TABLES	vii
LIST OF FIGURES	viii
LIST OF SYMBOLS	xi
ABSTRACT	xiv
CHAPTERS:	
I. INTRODUCTION	1
II. TWO LEVEL OPTIMIZATION METHOD. DUAL GAPS AND THEIR RESOLUTION USING METHOD OF MULTIPLIERS ...	5
II.1. Review of the Previous Works on the Non- convex Optimization	5
II.2. Statement of the Problem and the Two-Level Procedure	7
II.3. Dual Gaps	10
II.4. Resolution of Dual Gaps Using the Method of Multipliers	13
II.5. Computational Separability	22
II.6. The Algorithm	24
II.7. A Discrete Minimum Principle	26
II.8. Discussion	34
III. A STRONG VERSION OF THE DISCRETE MINIMUM PRINCIPLE	37
III.1. Review of Previous Works	37
III.2. Statement of the Problem and Its Lagrangian Formulation	39

TABLE OF CONTENTS (continued)

	<u>Page</u>
III.3. Development of a Stronger Version of the Discrete Minimum Principle	45
III.4. The Algorithm and Computational Characteristics	51
III.5. Discussion	53
IV. EXAMPLES OF NONCONVEX OPTIMIZATION	56
IV.1. Numerical Examples	56
IV.1.a. Two-stage examples	56
IV.1.b. Three-stage example with recycle	60
IV.1.c. Soiand's example	64
IV.1.d. Jackson and Horn's counter- example	64
IV.1.e. Denn's counterexample	69
IV.2. The Design of a Heat Exchange Network ...	72
V. SYNTHESIS OF PROCESS FLOWSHEETS. A GENERAL REVIEW	83
VI. BRANCH AND BOUND STRATEGY FOR THE SYNTHESIS OF OPTIMAL SEPARATION SCHEMES	90
VI.1. Previous Works on the Synthesis of Separation Schemes	90
VI.2. Statement of the Problem and the List Techniques for the Representation of the Separation Operations.....	93
VI.3. Branch and Bound Strategy	100
VI.4. Examples	107
VI.4.a. Example 1: n-butylene purification system	108
VI.4.b. Example 2: olefins-paraffins separation system	121

TABLE OF CONTENTS (continued)

	Page
VI.5. Discussion	132
VII. EVOLUTIONARY SYNTHESIS OF PROCESS FLOWSHEETS ...	134
VII.1. A General Philosophy on Evolutionary Synthesis	134
VII.2. Evolutionary Synthesis of Optimal Multi- component Separation Sequences	143
VII.2.a. Representation of separation sequences as binary trees	143
VII.2.b. Neighboring flowsheets and the evolutionary rules for a separation sequence	146
VII.2.c. Polish strings and their repre- sentation of separation sequences	150
VII.2.d. Proof of the completeness of the evolutionary rules	155
VII.2.e. Evolutionary strategy	161
VII.3. Examples of Evolutionary Synthesis .	163
VII.3.a. Example 1: synthesis of a solids' separation system	163
VII.3.b. Example 2: synthesis of a multicomponent separation sequence	167
VII.4. Discussion	174
VIII. CONCLUSIONS AND RECOMMENDATIONS FOR FURTHER RESEARCH	178
APPENDICES	183
APPENDIX A	184
BIBLIOGRAPHY	187
BIOGRAPHICAL SKETCH	191

LIST OF TABLES

<u>Table</u>		<u>Page</u>
1	Points Generated for the Two-Stage Example	58
2	Points Generated for the Modified Two-Stage Example	61
3	Points Generated for Soiland's Example	65
4	The Numbers of Distinct Separators $B(N)$ and Distinct Flowsheets $F(N)$ for a Mixture of N Components and One Separation Method	96
5	Initial Feed to the n-Butylene Purification System	109
6	Generated Flowsheets and Their Minimum Costs	120
7	Specifications of the Initial Feed and the Desired Products for Example 2	122
8	Specification of the Solids in Example 1	165
9	The Evolutionary Steps Taken During the Synthesis of the Solids' Separation System	168
10	Table Indicating the Effectiveness of the Proposed Evolutionary Strategy in Synthesizing an Optimal Separation Sequence (One Separation Method Used)..	176
11	Table Indicating the Effectiveness of the Proposed Evolutionary Strategy in Synthesizing an Optimal Separation Sequence (Three Separation Methods Used)	177

LIST OF FIGURES

<u>Figure</u>		<u>Page</u>
1	Geometric View of the Success of the Dual Approach ..	12
2	Geometric View of the Failure of the Dual Approach to Yield the Primal Solution	14
3	Improvement of the Dual Bound $h^*(\tilde{\lambda})$ for the Penalized Problem Over the Dual Bound $h(\lambda)$ for the Unpenalized Problem for the Same Multipliers $\tilde{\lambda}$	36
4	Relationship of Penalty Constant K Required for Three Lagrangian Based Optimization Algorithms	55
5	Two-stage Example	57
6	Three-stage Example with Recycle	62
7	Jackson and Horn's Counterexample	66
8	Heat Recovery Process	74
9	Uncoupled Heat Recovery Process Showing the Enthalphy Variables Used in the Two-level Optimization Method.	75
10	Diagram Showing Notation for a Single Exchanger	78
11	All the Distinct Separators Generated and the Basic Flowsheet for a Fictitious Example of 4 Components Using 2 Separation Methods	98
12	Two-stage Example to Demonstrate the Insensitivity of the Dual Function to the Values of the Lagrange Multipliers	106
13	Generation of the Basic Flowsheet for the Synthesis of the n-Butylene Purification System	111
14	Generation of all the Flowsheets Starting With Separator 1 for the n-Butylene Purification System Example	112

LIST OF FIGURES (continued)

<u>Figure</u>		<u>Page</u>
15	Generation of all the Flowsheets Starting with Separators 2 or 5 for the n-Butylene Purification System Example	114
16	All the Distinct Separators Employed During the Synthesis by Branch and Bound of the n-Butylene Purification System	116
17	Nearly Optimum Flowsheets Retained at the End of the Branch and Bound Synthesis of the n-Butylene Purification System	119
18	All the Distinct Separators Employed During the Synthesis by Branch and Bound of the Olefins, Paraffins Separation System	123
19	Generation of the Basic Flowsheet for the Synthesis of the Olefins, Paraffins Separation System	127
20	Nearly Optimum Flowsheets Retained at the End of the Branch and Bound Synthesis of the Olefins, Paraffins Separation System	128
21	Generation of all Flowsheets Starting with Separator 1 for the Olefins, Paraffins Separation System	129
22	Generation of all Flowsheets Starting with Separators 2 or 3 for the Olefins, Paraffins Separation System	130
23	Generation of all Flowsheets Starting with Separator 4 for the Olefins, Paraffins Separation System	131
24	An Illustrative Diagram Showing Two Alternate Sets of Evolutionary Rules for a Family of Flowsheets A_i	137
25	Flowsheet A and Its Neighboring Flowsheets B,C,D and E Resulting from A with Simple Structural Modifications	139
26	A Separation Sequence (A), Its Corresponding Binary Tree (B) and the Skeleton Structure (C) Corresponding to This Tree	144

LIST OF FIGURES (continued)

<u>Figure</u>		<u>Page</u>
27	Flowsheet (A), a Down Neighbor (B) to It, and a New Separator Type Neighbor (C) to It	147
28	Separation Sequences Corresponding to the Binary Trees of Figures 27 A and B	149
29	A Binary Tree for the Algebraic Expression $3x^2y + z^2/u$	151
30	Interpreting a Polish String - Development of an Algebraic Expression Using a Stack of Operands and a Set of Operators	153
31	The Operators of a Polish String and Their Operands, and the Schematic Representation of the Generation of the Neighboring Flowsheets by Applying Rules 1 and 2	156
32	Flowsheet A-1 and Its Neighboring Flowsheets A-2, A-3 and A-4 Generated Using Rules 1 and 2	158
33	A Schematic Representation of a System for the Separation of Solids and the Corresponding Binary Tree	166
34	Flowsheets Generated During the Evolutionary Synthesis of the n-Butylene Purification System Starting from Flowsheet (a)	170
35	The Evolution Process for the n-Butylene Purification System Starting from the Flowsheet (a)	171
36	The Evolution Process for the n-Butylene Purification System Starting from the Flowsheet (k)	172
37	Flowsheets Generated During the Evolutionary Synthesis of the n-Butylene Purification System Starting from Flowsheet (k)	173

LIST OF SYMBOLS

A_i	= Heat transfer area of the i -th heat exchanger.
a, b, c	= Flowrates of hot streams in a heat exchange network.
C_i	= Heat capacity of stream i .
c_i	= Cost of the i -th heat exchanger.
D	= Set of feasible dual variables or multipliers.
F	= Scalar return function.
F^*	= Scalar return function for the augmented problem, may be subscripted.
f_i	= Stage transformation function, for stage i .
f_j	= Scalar return function for subsystem j .
g	= Vector of equality constraint functions, may be subscripted.
H	= Supporting hyperplane for the set R .
H_i	= Hessian matrix of subsystem i .
H_i	= Hamiltonian function. Subscripted with i refers to the Hamiltonian for the subsystem i .
H^*	= Hamiltonian function for the augmented problem.
h	= Dual function.
h_j	= Vector of inequality constraint functions for subsystem j .
$I(j)$	= The set of i such that stream i is an input to subsystem j .
K	= Penalty constant, always nonnegative.
L	= Lagrangian function.
L^*	= Lagrangian function for the augmented problem, may be subscripted.

ξ_i	= Sub-lagrangian for subsystem i.
ξ_i^*	= Sub-lagrangian for subsystem i after the linear approximation of nonseparable terms.
$O(j)$	= The set of i such that stream i is an output of subsystem j.
$P(q)$	= Scalar quantity = $g^T(q)g(q)$.
Q	= Heat duty of an exchanger.
q	= Composite vector variable ($x u$), may be subscripted.
R	= The set (z_0, z) such that $z_0 \geq w(z)$, $z \in Z$, may be subscripted.
S	= Constrained variable set, may be subscripted.
s	= Direction vector, may be subscripted.
T_o, T_p	= Temperatures of a cold stream at the entrance and the exit of a heat exchange network.
t_i	= Stage transformation function, for stage i.
t_a, t_b, t_c	= Temperatures at which the hot streams a,b,c are available.
u	= System decision variable vector, may be subscripted.
$w(z)$	= Minimum $\{F(x); g(x) = z, x \in S\}$.
$w^*(z)$	= Minimum $\{F^*(x); g(x) = z, x \in S\}$.
x	= Vector variable associated with interconnecting streams, may be subscripted.
y	= Vector variable associated with streams leaving the system, may be subscripted.
Z	= The set of z such that $\exists x \in S \ni g(x) = z$.
z	= Perturbation vector, may be subscripted.

Greek Letters

α	= Scalar parameter.
α_i	= Positive constant, usually = 0.6.
β	= Scalar parameter.

γ_i	= Positive constant.
ΔT_m	= Log mean temperature difference.
ϵ	= Small positive quantity.
θ	= System decision variable vector, may be subscripted.
λ	= Vector of Lagrange multipliers, may be subscripted.
μ	= Vector of Lagrange multipliers for the augmented problem.
v	= Vector of Kuhn-Tucker multipliers, may be subscripted.
π	= Scalar parameter.

Mathematical Symbols

\exists	= Such that.
\exists, \nexists	= There exists, there does not exist.
\forall	= For all.
\emptyset	= Zero vector.
\subset	= Subset of.
\cap	= Intersection with.
\cup	= Union with.

Abstract of Dissertation Presented to the Graduate Council
of the University of Florida in Partial Fulfillment
of the Requirements for the Degree of Doctor of Philosophy

OPTIMIZATION OF NONCONVEX SYSTEMS
AND THE SYNTHESIS OF OPTIMUM
PROCESS FLOWSHEETS

by

George Stephanopoulos

June, 1974

Chairman: Arthur W. Westerberg

Major Department: Chemical Engineering

Previous efforts in the area of optimization of chemical processes have not accounted for the nonconvexities commonly encountered in such systems. These nonconvexities cause many of the proposed large scale optimization strategies to fail. The subject also of the chemical process synthesis was largely ignored until recently despite its importance in the chemical engineering practice. This dissertation presents techniques to overcome the deficiencies of two related and often studied optimization methods in the presence of nonconvexities and develops strategies for the synthesis of process flowsheets.

Many Lagrangian based methods for the optimization of large scale systems require that the Lagrangian function possess a saddle point. The two-level optimization method may not be generally applicable to chemical process design problems due to the mathematical character of commonly encountered constraint sets and objective functions, which commonly do not allow the existence of saddle points for the Lagrangian function. The dissertation presents a method to overcome these

shortcomings of the two-level optimization procedure by employing Hestenes' method of multipliers. The objective function is augmented by a penalty term which is the sum of the squares of the connection constraints multiplied by a positive constant. This penalty term under certain conditions turns every stationary point of the Lagrangian into a saddle point thus securing the success of the two-level method. The separability of the initial system which was lost because of the penalty term is regained by expanding the nonseparable terms into a Taylor Series and retaining only the linear part of it. As a direct extension of the above strategy the dissertation presents the development of a stronger version of the discrete minimum principle. Two algorithms have been developed to implement these theoretical results. The success of the new methods has been demonstrated in several small size numerical examples and in the design of a simple heat recovery network on which the previous methods fail.

The dissertation also develops two strategies for the synthesis of chemical process flowsheets. The first is a branch and bound strategy which exploits the bounding properties of the dual and primal values which can be obtained for the flowsheet objective function. The flowsheets are constructed in a block-by-block building procedure, and this method was used to synthesize optimal multicomponent separation sequences. In the method list processing techniques are used to develop the distinct separators which can occur. At the end of the synthesis a small number of nearly optimum flowsheets is retained. Further screening among them is possible to locate the optimum

flowsheet. This strategy is demonstrated in two different examples with very encouraging results.

The second synthesis strategy, the evolutionary strategy, is considered next and is systematized for use in the synthesis of general process flowsheets. The evolutionary synthesis procedure is broken into four subtasks: a) the development of a starting flowsheet, b) the generation of evolutionary rules to produce the structural modifications to be considered during the synthesis, c) the developing of the proper evolutionary strategy to lead to the optimum solution in the most effective manner, and d) the screening among the current flowsheet and the alternative flowsheets generated. Notions such as that of the neighboring flowsheets and the evolutionary strategy help to put evolutionary synthesis in a correct perspective. The evolutionary approach is illustrated with the synthesis of two distinct separation problems with very encouraging results.

CHAPTER I

INTRODUCTION

The essential concern, purpose, and culmination of engineering is design. In chemical engineering this is the prevailing and most important factor of its scope. Very often, a final design is achieved without due consideration to all aspects of the design morphology. This is necessitated by the complexity of the design problem and the state of the limited engineering advances in certain areas. Proper design procedure includes the three essential stages of synthesis, analysis and evaluation. The design process is complicated by the interrelationships existing among these stages. Frequently these interrelationships are complex and cause design to be an iterative process, requiring the special attention of the designer and the development of flexible strategies which will lead to good solutions.

Analysis is a term which is equally familiar to both practicing engineers and students of engineering. It has been developed deductively and quantitatively to a high degree. Strategies have been developed to analyze whole processes and very effective, sophisticated methods have been proposed to resolve the complicated, time consuming activities of the design process.

Considerable work has been done and is in progress on optimization theory for large structured systems. Both theory and applications have found very fertile ground in chemical engineering. The particular

feature of chemical processes (i.e., sparseness and complexity) have caused the development of highly effective strategies for analysis and optimization. Chemical process design has been the instigator of such developments.

The synthesis stage of the design procedure was largely ignored until recently in the chemical engineering literature despite its importance in chemical engineering practice. During the last few years the importance of creativity, innovation and invention in designing chemical processes has been stressed and has received the proper attention.

Because synthesis is such an important step, it became the principal goal of this thesis. Initially we were exploring the use of McGalliard's (McGalliard, 1971) approach to structural synthesis for the synthesis of optimal multicomponent separation schemes. This strategy involves the development of dual bounds for alternate flowsheets. Thus the first problem of concern was a good initial estimation of the Lagrange multipliers to be used for the evaluation of dual bounds. A more thorough and detailed investigation of the physical meaning of the Lagrange multipliers and the discovery of Hestenes' method in the literature (Hestenes , 1969) led to the development of a new method to overcome the deficiencies of Lasdon's (Lasdon, 1964) two-level optimization method.

Then it became evident that a stronger version of the discrete minimum principle could also be established. Thus two new strategies evolved which can very likely be used effectively (as their application to several small examples has indicated) to optimize large-scale systems in the presence of nonconvexities. These approaches are of

particular interest to a chemical engineer, since the cost functions used in process design involve the throughput to a unit raised to the 0.6 power which is a characteristic nonconvex function.

Furthermore we explored the use of the bounding properties of the primal and dual functions in connection with list processing techniques to generate very good alternate solutions to the multi-component separation problem. Finally, the evolutionary approach to the synthesis of optimal process flowsheets was systematized, and all these principles and ideas were illustrated in the synthesis of an optimal separation scheme.

Thus, in Chapter II we discuss the two-level optimization method, its theoretical foundations, its advantages and its drawbacks. The generation of dual gaps because of nonconvexities and their resolution using Hestenes' method of multipliers is discussed. An algorithmic procedure employing these ideas is described. In Chapter III the "classical" discrete minimum principle is outlined, and its shortcomings because of nonconvexities are defined. Again Hestenes' multipliers are used to develop a strong version of the discrete minimum principle, and a new algorithmic procedure is proposed. In Chapter IV the theories developed in Chapters II and III are tested on some simple numerical examples and also some examples drawn from the chemical engineering literature. In Chapter V a general review is presented of previous works in the area of process flowsheet synthesis. Then a branch and bound strategy, using list processing techniques, is outlined for the synthesis of optimal multicomponent separation sequences in Chapter VI, while Chapter VII develops a systematic evolutionary approach for the synthesis of process flowsheets.

Chapter VII concludes with an illustrative presentation of the principles governing the evolution of the design on a separation problem.

Finally in Chapter VIII we summarize the results of this thesis and outline a program for further related research in the area of chemical process design.

CHAPTER II

TWO-LEVEL OPTIMIZATION METHOD. DUAL GAPS. RESOLUTION USING METHOD OF MULTIPLIERS

II.1. Review of Previous Works on the Nonconvex Optimization.

The Lagrangian approach, representing a dual method, has often been proposed for solving optimization problems. In many engineering design problems, the system to be optimized comprises several fairly complex subsystems or units which are connected together by sets of equality constraints. By appending the equality constraints to the objective function with Lagrange multipliers, the Lagrange function, for fixed multiplier values, is separable, leaving a subproblem for each subsystem. The approach is therefore very attractive for this type of problem. It can fail however because of the presence of a dual gap at the solution point (Everett, 1963; Lasdon, 1970). Unfortunately this failure is common in engineering design problems; therefore, the development of a method to resolve dual gaps is of great importance.

Gaps may arise for various reasons, and a quite thorough treatment of their causes and resolutions is presented by Greenberg (1969). He reviewed the use of nonlinear supports (Gould, 1969; Bellman-Karush, 1961), the use of surrogates (Greenberg-Pierskalla, 1970; Glover, 1968), the use of cuts via dominance and efficiency concepts (Loane, 1971; Greenberg, 1969), and the use of branch and bound method designed for finite separable problems. It should be noted that in all these methods to resolve gaps, inequality connection constraints are emphasized.

Bellmore, Greenberg and Jarvis (1970) have examined the use of penalty functions to resolve gaps, and they reviewed the relationship between the original problem and the augmented one along with proposed solution procedures.

Falk and Soland (1969) have proposed an algorithm to solve the separable nonconvex programming problems where the constraints are only upper and lower bounds on the variables. Soland (1971) has extended the algorithm to include inequality constraints of a more general form. Both the algorithms are of a branch and bound type and solve a sequence of problems with convex objective functions. These problems correspond to successive partitions of the feasible set. Greenberg (1973), in a recent publication, has provided a sharper lower bound on the optimum value with less computation using the Generalized Lagrange Multiplier method, and the Falk-Soland algorithm can be modified appropriately.

All the above methods have a major drawback: the separability of the objective function and the constraints, if it existed in the original problem, is destroyed after the proposed modifications. Only the last method reported by Greenberg (1969) and using a branch and bound technique preserves the separability, but it is only applicable for finite problems. Separability is a characteristic providing many advantages for the solution of a large system, and it is very desirable to preserve it.

In applying the Lagrangian approach and structural sensitivity analysis (McGalliard, Westerberg ,1972) in engineering design problems, we want to resolve the dual gap problem by keeping the structural characteristics of the system.

In the present work an algorithm is developed which makes use of the penalty function approach together with a linear approximation of the nonseparable terms. The original problem is replaced by a sequence of problems, each one yielding a tighter dual bound on the optimum solution. The solutions to the above problems form a nondecreasing sequence of real numbers, bounded from above. Under certain conditions developed later on, this sequence of solutions converges to the solution of the original problem. The structural characteristics of the original system are preserved: each subproblem is solved separately.

II.2. Statement of the Problem and the Two-Level Procedure

As a basis for the description of the two-level optimization procedure, of the encountered dual "gaps", and of their proposed resolution, we will consider the following model which represents many engineering system design problems (as opposed to resource allocation problems):

$$I(j) = \{i ; \text{stream } i \text{ is an input to subsystem } j\}$$

$$O(j) = \{i ; \text{stream } i \text{ is an output of subsystem } j\}$$

x_i is a vector variable associated with the interconnecting stream i .

u_j is a vector decision variable associated with subsystem j .

Define vector variable q_u as

$$q_j = (x_{k_1}, \dots, x_{k_{n_j}} | u_j), \quad k_i \in I(j). \quad (\text{II-1})$$

The transformation equations which connect the subsystems are

$$x_i = t_i(q_j), \quad i \in O(j), \quad j = 1, \dots, n. \quad (II-2)$$

The overall system return function is the sum of the subsystem return functions:

$$F = \sum_{j=1}^n f_j(q_j). \quad (II-3)$$

The system constraint set, excepting the interconnection constraints (II-2), is separable, i.e.,

$$q_j \in S_j = \{q_j \mid h_j(q_j) \leq 0\} \quad j = 1, \dots, n.$$

Thus, the overall optimization problem is

$$\text{Minimize } F = \sum_{j=1}^n f_j(q_j) \quad (II-4)$$

$$\text{subject to } x_i = t_i(q_j), \quad i \in O(j), \quad j = 1, \dots, n$$

$$q_j \in S_j, \quad j = 1, \dots, n.$$

The Lagrange function for the above problem described by (II-4) is given by

$$L = \sum_{j=1}^n f_j(q_j) + \sum_{j=1}^n \sum_{i \in O(j)} \lambda_i^T (t_i(q_j) - x_i), \quad (II-5)$$

where the λ_i are Lagrange multipliers. Rearranging the terms in (II-5) the Lagrangian can be written as follows:

$$L = \sum_{j=1}^n \{ f_j(q_j) + \sum_{i \in O(j)} \lambda_i t_i(q_j) - \sum_{i \in I(j)} \lambda_i x_i \} = \sum_{j=1}^n \ell_j(q_j; \lambda). \quad (II-6)$$

For fixed λ the problem of minimizing L becomes separable, and it is equivalent to solving the subproblems

$$\begin{aligned} & \text{Minimize } \ell_j(q_j; \lambda) \\ & j=1, \dots, n \end{aligned} \quad (II-7)$$

$$\text{subject to } q_u \in S_j.$$

Let us now define a dual function and a dual problem. The dual function is

$$h(\lambda) = \sum_{j=1}^n (\min_{q_j \in S_j} \ell_j(q_j; \lambda)), \quad (II-8)$$

and the dual problem is

$$\begin{aligned} & \text{Maximize } h(\lambda) \\ & \end{aligned} \quad (II-9)$$

$$\text{subject to } \lambda \in D, \quad D = \{\lambda; h(\lambda) \text{ exists}\}.$$

The two-level optimization procedure requires the following two distinct operations:

First level: Calculate $h(\lambda)$ by solving the n subproblems (II-7)

Second level: Adjust the multipliers, λ , so as to satisfy the interconnection constraints in (II-4)

In effect this procedure solves the dual problem described by (II-9).

The important question concerning this procedure relates to the existence of a saddle point for the Lagrange function of the problem. The following theorems provide the theoretical basis and give some answers to the saddle point existence question. For further details and proofs of the theorems, see Lasdon (1970).

Theorem 1: Let $\lambda^0 \in E^n$. A point (q^0, λ^0) is a constrained saddle point for $L(q, \lambda)$ if and only if 1) q^0 minimizes $L(q, \lambda^0)$ over S and 2) $x_i - t_i(q_j^0) = 0$, $i \in O(j)$, $j = 1, \dots, n$.

Theorem 2: If (q^0, λ^0) is a saddle point for L , then q^0 solves the primal problem described by (II-4).

Theorem 3: $h(\lambda)$ is concave over any convex set of D .

Theorem 4: $h(\lambda) \leq F(q)$ for all $q \in S$ such that $x_i - t_i(q_j) = 0$, $i \in O(j)$, $j = 1, \dots, n$ and for all $\lambda \in D$.

II.3. Dual Gaps

The basic drawback of the Lagrangian approach, and therefore of the two-level optimization procedure, is its failure to find the solution of a problem when the solution is in a "dual gap".

To provide additional insight into the relationship between the primal and the dual problems and to demonstrate the formation of gaps, we will give at first a geometric interpretation of the procedure and second the theoretical justification of the failure of the two-level approach in certain cases (Lasdon, 1970 ; Rockafellar, 1967).

Consider the family of the perturbed problems, with perturbations z_i :

$$\text{Minimize } F(q) = \sum_{j=1}^n f_j(q_j)$$

$$\text{subject to } x_i - t_i(q_j) = z_i, \quad i \in O(j), \quad j=1, \dots, n$$

$$q_j \in S_j \quad j=1, \dots, n$$

The primal problem corresponds to $z_i = 0$, $i \in O(j)$, $j=1, \dots, n$. Assuming continuity of the objective function and of the constraint functions, let us define

$$w(z) = \text{minimum } \{F(q); x_i - t_i(q_j) = z_i, i \in O(j), j=1, \dots, n, q \in S\}$$

$$\text{where } z^T = [z_i^T, \text{ for all } i \in O(j), j=1, \dots, n].$$

The domain of w is

$$Z = \{z; \text{ there exists a } q \in S \text{ such that } x_i - t_i(q_j) = z_i, i \in O(j), j=1, \dots, n\}.$$

Consider now the set $R \subset E^{m+1}$

$$R = \{(z_0, z); z_0 \geq w(z), z \in Z\}.$$

If w and Z are convex then R is convex. We shall call this space, containing R , the WZ space.

The following theorem demonstrates the connection between duality and supporting hyperplanes for the set R , see Lasdon (1970) in WZ.

Theorem 5: If $\hat{q} \in S$ and $\hat{\lambda} \in E^m$, then \hat{q} minimizes $L(q, \lambda)$ over S , if and only if $H = \{(z_0, z); z_0 - \lambda z = L(\hat{q}, \hat{\lambda})\}$ is a supporting hyperplane for the set R at the point, $(F(\hat{q}), x_i - t_i(\hat{q}_j) = z_i, i \in O(j), j=1, \dots, n)$.

Supporting hyperplanes exist for every point and therefore for the solution point, if R is a convex set (Fig. 1). In the case of

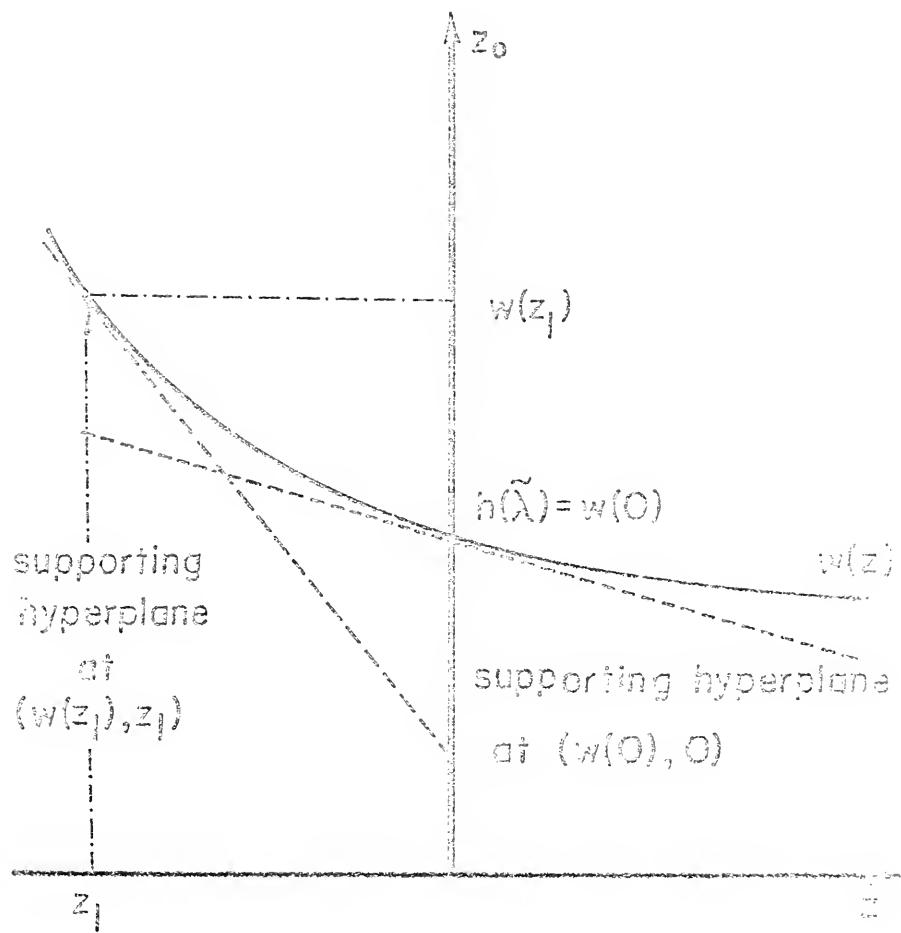


Figure 1. Geometric View of the Success of the Dual Approach

nonconvex R sets, there are regions consisting of constraint vectors that are not generated by any vector λ (Fig. 2). Optimum solutions for constraints inside such inaccessible regions cannot be discovered by straightforward application of the two-level optimization procedure, and must be sought by other means.

The following corollary (Greenberg, 1969) helps to anticipate the existence of such an inaccessible region, which will be termed a "dual gap".

Corollary: A dual gap arises if some choice of Lagrange multipliers $\lambda \in D$ produces at least two solutions to the Lagrangian problem (II-8) with distinct $z_i = x_i - t_i(q_j) \neq 0$, $i \in O(j)$, $j=1, \dots, n$.

II.4. Resolution of Dual Gaps Using the Method of Multipliers

Let us assume that the solution to problem (II-4) is in a dual gap. According to Theorem 5, the two-level approach fails to find the solution. The method of multipliers developed by Hestenes (1969) will be used to cure this shortcoming of the two-level method.

Let us make the following notational changes:

$$g_{ij} = x_i - t_i(q_j), \quad i \in O(j), \quad j=1, \dots, n$$

$$g_j^T = [g_{i_1j}^T, g_{i_2j}^T, \dots, g_{i_mj}^T]; \quad j=1, \dots, n; \quad i_k \in O(j); \quad k=1, \dots, m$$

$$g^T = [g_1^T, g_2^T, \dots, g_n^T]$$

$$q^T = [q_1^T, q_2^T, \dots, q_n^T]$$

$$h^T = [h_1^T, h_2^T, \dots, h_n^T].$$

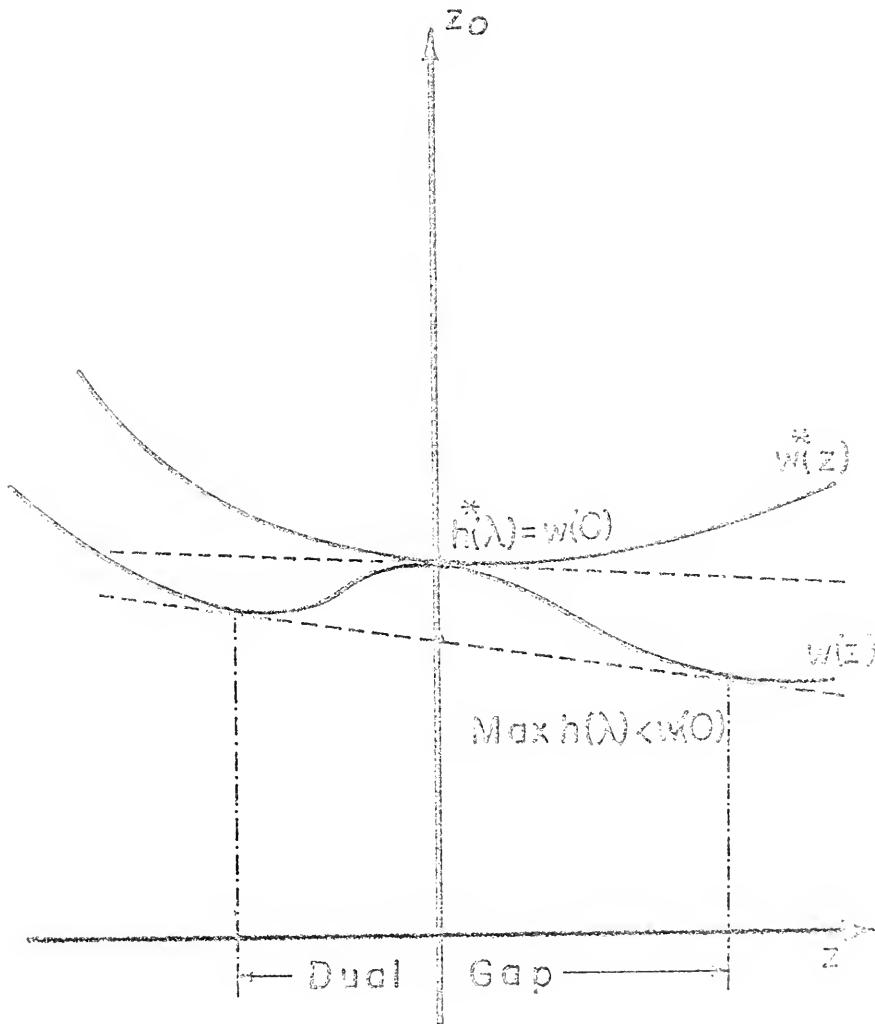


Figure 2. Geometric View of the Failure of the Dual Approach to Yield the Primal Solution

Then problem (II-4) is written as follows:

$$\begin{aligned} \text{Min } F = & \sum_{j=1}^n f_j(q_j) = F(q) \\ \text{s.t. } & g(q) = 0 \\ & q \in S \end{aligned} \tag{II-P1}$$

where $S = S_1 \times S_2 \times \dots \times S_n$.

Consider now the following augmented problem:

$$\begin{aligned} \text{Min } F_i^*(q, k_i) = & F(q) + K_i g^T(q)g(q) \\ \text{s.t. } & g(q) = 0 \\ & q \in S \\ & K_i > 0. \end{aligned} \tag{II-P2}$$

Let us now examine the relationship between problems (II-P1) and (II-P2).

Let us make the following assumptions:

- A-1: Objective function $F(q)$ and constraints $g(q)$ are of class C^2 .
- A-2: The set S is compact and nonempty.

Theorem 6 (Hestenes, 1969): Let q_0 be a nonsingular solution to the problem (II-P1), then there exists a multiplier vector λ and a constant K_i such that the point q_0 is an unconstrained minimum of the Lagrangian of problem (II-P2).

Proof: Since q_0 is the solution of the problem (II-P1), the following conditions are true:

$$g(q_0) = 0, \quad \left. \frac{\partial L}{\partial q} \right|_{q_0} = 0, \text{ where } L = F(q) - \lambda^T g(q) - v^T h(q)$$

$$\text{and } \Delta L = \left. \Delta q^T \frac{\partial^2 L}{\partial q^2} \right|_{q_0} \Delta q > 0 \text{ for all permissible variations } \Delta q,$$

such that

$$\Delta g = \left. \frac{\partial g}{\partial q^T} \right|_{q_0} \Delta q = 0 \text{ and } v_j \left[\left. \frac{\partial h_j}{\partial q_j} \right|_{q_0} \Delta q_j \right] \geq 0 \quad j=1,2,\dots,n.$$

Note: The Kuhn-Tucker multipliers v_j are always nonpositive at the solution to (II-P1). The Lagrangian of the problem (II-P2) is:

$$L_i^* = L + K_i g^T(q) g(q).$$

At the point q_0 :

$$\left. \frac{\partial L_i^*}{\partial q} \right|_{q_0} = \left. \frac{\partial L}{\partial q} \right|_{q_0} = 0$$

and

$$\Delta L_i^* = \Delta L + K_i \left[\left. \frac{\partial g}{\partial q^T} \right|_{q_0} \Delta q \right]^T \left[\left. \frac{\partial g}{\partial q^T} \right|_{q_0} \Delta q \right] \geq 0.$$

For permissible variations Δq such that $\Delta g = 0$ and $v_j \Delta h_j \geq 0$, $\Delta L_i^* > 0$ and for nonpermissible variations Δq , it is possible that $\Delta L < 0$, but there exists a $K_i > 0$ such that

$$\Delta L_i^* > 0$$

and the proof is complete.

From Theorem 6 we conclude that there exist a vector λ and a $K_i > 0$

such that the point (q_0, λ) is a saddle point for the Lagrangian L_i^* , of the augmented problem (II-P2). Therefore the problem (II-P1) can be solved using the two-level method, if it is augmented by the penalty term $K_i g^T(q)g(q)$. Define:

$$w(z) = \min_{q \in S} \{F(q) : g(q) = z\}$$

$$w_i^*(z) = \min_{q \in S} \{F_i^*(q, K_i) : g(q) = z\} \quad .$$

Let us examine now how the value of K_i affects the $w_i^*(z)$ and under what conditions a proper choice of K_i yields a supporting hyperplane for the set R_i , where

$$R_i = \{(z_0, z) : z_0 \geq w_i^*(z), z \in Z\}$$

at the solution point $(w_i^*(0), 0)$. The following theoretical treatment will demonstrate the effect of the penalty term on the resolution of dual gaps via a geometrical representation.

Lemma 1: $w(z) < w_i^*(z)$ for $z \neq 0$ and $w(0) = w_i^*(0)$. More generally, if $K_j > K_i$ then $w_j^*(z) > w_i^*(z)$ and $w_j^*(0) = w_i^*(0)$.

Proof: The equality for $z = 0$ is direct since the penalty term is then zero. Let \hat{q} minimize $F(q)$ with $g(\hat{q}) = z$, then \hat{q} minimizes F_i^* with $g(\hat{q}) = z$ because if this is not true and \tilde{q} minimizes F_i^* with $g(\tilde{q}) = z$, then

$$F(\tilde{q}) + K_i g^T(\tilde{q})g(\tilde{q}) < F(\hat{q}) + K_i g^T(\hat{q})g(\hat{q})$$

and consequently

$$F(\tilde{q}) < F(\hat{q})$$

which is not true since by assumption \hat{q} minimizes $F(q)$ with $g(\hat{q}) = z$.

Similarly:

$$\min_{q \in S} F_j^*(q, K_j) = \min_{q \in S} \{F(q) + K_j z^T z\}$$

$$> \min_{q \in S} \{F(q) + K_i z^T z\} = \min_{q \in S} F_i^*(q, K_i).$$

Therefore $w_j^*(z) > w_i^*(z)$ for $z \neq 0$ Q.E.D.

Lemma 1 implies that the curve $w(z)$ vs. z (in the one-dimensional case) is moved upwards as K_i increases, keeping always the same value at $z = 0$.

Lemma 2: If $w(z)$ is a continuous function with continuous first derivatives, $K_j > K_i$, and

$$H(K_\lambda) = \max_{\lambda} h_\lambda^*(\lambda) = \max_{\lambda} \min_{q \in S} L_\lambda^*(q, \lambda, K_\lambda) = \max_{\lambda} \min_{q \in S} [F_\lambda^* - \lambda^T g(q)]$$

then

$H(K_j) > H(K_i)$ and $H(K_j) = H(K_i)$ only at the solution of the problem (II-P1).

Proof: Let

$$H(K_j) = F_j^*(q_{(j)}^*) - \lambda_{(j)}^{*T} g(q_{(j)}^*)$$

$$H(K_i) = F_i^*(q_{(i)}^*) - \lambda_{(i)}^{*T} g(q_{(i)}^*).$$

Assume that $H(K_j) \leq H(K_i)$, then

$$F_j^*(q_{(j)}^*) - \lambda_{(j)}^{*T} g(q_{(j)}^*) \leq F_i^*(q_{(i)}^*) - \lambda_{(i)}^{*T} g(q_{(i)}^*),$$

but

$$F_j^*(q_{(j)}^*) - \lambda_{(j)}^{*T} g(q_{(j)}^*) \geq F_j^*(q_{(j)}^*) - \lambda_{(i)}^{*T} g(q_{(j)}^*);$$

therefore

$$F_j^*(q_{(j)}^*) - \lambda_{(i)}^{*T} g(q_{(j)}^*) \leq F_i^*(q_{(i)}^*) - \lambda_{(i)}^{*T} g(q_{(i)}^*).$$

The last inequality implies that the point $(F_j^*(q_{(j)}^*), g(q_{(j)}^*))$ lies below or on the hyperplane, defined by:

$$H = \{(z_0, z) | z_0 - \lambda_{(i)}^{*T} z = H(K_i)\}.$$

Let us first examine the case where it lies below the hyperplane. Then

$$(F_j^*(q_{(j)}^*), g(q_{(j)}^*)) \notin R_i$$

where

$$R_i = \{(z_0, z) | z_0 \geq w_i^*(z), z \in Z\}.$$

But,

$$(F_j^*(q_{(j)}^*), g(q_{(j)}^*)) \in R_j$$

where

$$R_j = \{(z_0, z) | z_0 \geq w_j^*(z), z \in Z\}.$$

Because of Lemma 1 $R_j \subsetneq R_i$ and $(R_i - R_j) \cap R_j = (w(0), 0)$. Thus

$$(F_j^*(q_{(j)}^*), g(q_{(j)}^*)) \in R_i$$

which contradicts the above result. Therefore $H(K_j) \not\subset H(K_i)$.

In the case that the point $(F_j^*(q_{(j)}^*), g(q_{(j)}^*))$ lies on the hyperplane, because

$$(R_i - R_j) \cap R_i = (w(0), 0)$$

we conclude that $H(K_j) = H(K_i)$ only at the solution of (II-P1).

Theorem 7: If $w(z)$ is finite for all $q \in S$ and is continuous with continuous first derivatives and existing finite one-sided directional second derivatives in any direction s , at $z = 0$, then there exists a K^* finite such that for all $K > K^*$, $H(K) = H(K^*)$ = primal solution.

Proof: Since $w(z)$ is continuous with continuous first derivatives at $z = 0$, there exists a hyperplane tangent to $w(z)$ at the point $(w(0), 0)$. This is not a supporting hyperplane for the set R , since there is no saddle point for the Lagrangian of the problem (II-P1) inside the dual gap. This hyperplane is described by:

$$z_0 - \hat{\lambda}^T z = w(0) \quad \text{with some } z_0 \geq w(z).$$

Now we have to find a K_j such that the above hyperplane is a supporting hyperplane for the set R_j , i.e.,

$$z_0 = w(0) + \hat{\lambda}^T z \leq w_j^*(z).$$

We note that

$$w_j^*(z) = \min_{q \in E} \{F(q) + K_j g^T(q)g(q) | g(q) = z\}$$

therefore,

$$w(z) + K_i z^T z - \hat{\lambda}^T z \geq w(0)$$

and there must be a

$$K_i \geq \frac{w(0) - w(z) + \hat{\lambda}^T z}{z^T z} = \alpha \quad \text{for all } z \in Z = \{z \mid \exists q \in S, g(q) = z\}.$$

For $z \neq 0$ and with $w(z) > -\infty$ for $z \in Z$, a finite value of K_i exists, say α^2 . The only potential problem might occur at $z = 0$. We can use L'Hospital's rule to find the limit as $z \rightarrow 0$ along any direction s , namely

$$\lim_{z \rightarrow 0} \alpha = \lim_{\beta \rightarrow 0} \left[\frac{w(0) - w(\beta s) + \hat{\lambda}^T (\beta s)}{(\beta s)^T (\beta s)} \right] = -\frac{1}{2} \frac{\partial^2 w(\beta s)}{\partial \beta^2} = \text{finite},$$

if this limit exists. If the limit does not exist, then the finiteness of the one-sided directional second derivatives in any direction s , at $z = 0$, yields:

$$\lim_{\beta_j \rightarrow 0} \alpha = \alpha_j = \text{finite, along the direction } s_j.$$

Therefore choose K_i , such that

$$K_i \geq \max [\alpha^2, \alpha_j \text{ for every direction } s_j] = \text{finite}.$$

Therefore there exists a finite $K^* = K_i$ such that the hyperplane tangent to the $w(z)$ at the point $(w(0), 0)$, with a slope $\hat{\lambda}$, is a supporting hyperplane for the set R_i . Therefore the dual solution $H(K^*)$ equals the primal solution.

Now, for any $K > K^*$, $H(K) \geq H(K^*)$ but it cannot be that $H(K) = H(K^*)$ since that leads to $H(K) >$ primal solution. Therefore $H(K) = H(K^+)$. If $\partial w(z)/\partial z$ is to be discontinuous and $w(z)$ is continuous at $z = 0$, then K^* has to be $+\infty$, and the dual approach fails to give a solution for the problem (II-P2), but as K^* increases, approaching $+\infty$, it produces tighter lower bounds on the primal solution. The method can fail altogether if $w(z)$ is discontinuous at $z = 0$.

II.5. Computational Separability

The previously described resolution of the dual gaps suffers from a serious drawback, namely, the separability, which existed in problem (II-P1), has been destroyed in problem (II-P2) because of the cross-product terms in the penalty term. Separability preserves the structural characteristics of a system and induces properties which are always desired in solving a large-scale problem such as many of those in engineering design.

In the following paragraphs an algorithm is proposed which resolves the dual gap by preserving a computational separability of the system.

Using the initial notation, consider the penalty term

$$\begin{aligned}
 K_m(g(q)-z)^T (g(q)-z) &= K_m \sum_{j=1}^n \sum_{i \in 0(j)} [t_i(q_j) - x_i - z_i]^T [t_i(q_j) - x_i - z_i] \\
 &= K_m \sum_{j=1}^n \sum_{i \in 0(j)} \sum_{r=1}^{s_i} [t_{ir}^2 + x_{ir}^2 + z_{ir}^2 - 2t_{ir}z_{ir} - 2t_{ir}x_{ir} \\
 &\quad + 2x_{ir}z_{ir}]
 \end{aligned}$$

where z is the deviation from satisfying the constraints. Under

appropriate assumptions it has been seen that for large enough K_m , the solution of the problem can occur at $z_i = 0$ for all $i \in O(j)$ and $j=1, \dots, n$.

Each term of the above triple summation consists of separable terms $t_{ir}^2(q_j)$, x_{ir}^2 , $-2t_{ir}(q_j)z_{ir}$, and $-2x_{ir}z_{ir}$, and a crossproduct $-2t_{ir}(q_j)x_{ir}$ which is not separable. Expand the crossproduct in a Taylor Series and consider the following linear approximation around the point, \hat{x}_{ir} , $\hat{t}_{ir} = t_{ir}(\hat{q}_j)$:

$$t_{ir}(q_j)x_{ir} \approx -\hat{t}_{ir}\hat{x}_{ir} + \hat{t}_{ir}x_{ir} + t_{ir}\hat{x}_{ir}.$$

Then the Lagrangian of the problem (II-P2) with constraints $g(q) = z$ (instead of $g(q) = 0$) and Lagrange multipliers μ takes the following approximate separable form:

$$\begin{aligned} L_m^* &\approx \sum_{j=1}^n \left[f_j(q_j) + K_m \sum_{i \in O(j)} \sum_{r=1}^{s_i} [t_{ir}^2 + x_{ir}^2 - 2t_{ir}z_{ir} + 2\hat{t}_{ir}\hat{x}_{ir} - 2t_{ir}\hat{x}_{ir}] \right. \\ &\quad + K_m \sum_{i \in I(j)} \sum_{r=1}^{r_i} [x_{ir}^2 - 2\hat{t}_{ir}x_{ir} - 2x_{ir}z_{ir}] \\ &\quad \left. - \sum_{i \in O(j)} \mu_i^T(t_i - z_i) + \sum_{i \in I(j)} \mu_i^T x_i \right] \\ &= \sum_{j=1}^n \lambda_{mj}^*(q_j, \mu). \end{aligned}$$

Thus the problem of minimizing this form for L_m^* for fixed K_m and μ is equivalent to solving the subproblems:

$$\min_{q_j} \ell_{m_j}^*(q_j, \mu)$$

$j=1, \dots, n$

$$\text{s.t. } q_j \in S_j.$$

II.6. The Algorithm

Using the multiplier method (Section II-4) along with the above-mentioned linear approximation for the crossproduct terms, the following algorithmic procedure is developed which resolves the dual gap by preserving a computational separability of a large-scale system. In Section II-7 a minimum principle is developed which completes the theoretical foundation on which the algorithm is based.

Step 1: Assume a value for the penalty constant K_m .

Step 2: Assume values for Lagrange multipliers λ (we shall relate these to μ in the step 4).

Step 3: Assume a point $(\hat{x}_i, t_i(\hat{q}_i))$ for each $i \in O(j)$ and $j=1, 2, \dots, n$.

Step 4: Put $z_i = t_i(\hat{q}_j) - \hat{x}_i$ and $\mu_i = 2K_m z_i + \lambda_i$. Form the subproblems based on the linear approximation for the crossproduct term.

Step 5: Find \tilde{q}_j for $j=1, 2, \dots, n$ which solves

$$\min_{q_j \in S_j} \ell_{m_j}^*.$$

Step 6: Update \hat{x}_i , $t_i(\hat{q}_j)$ and iterate from step 4 until $z_i = t_i(\tilde{q}_j) - \tilde{x}_i$.

Step 7: Update the Lagrange multipliers λ and go to step 3 until the

$\max_{\mu} \min_{q \in S} L_m^*$ has been attained. Check the constraints. If

satisfied stop, otherwise update K_m and go to step 2.

The algorithm described above requires that a sequence of Max-Min problems be solved, each possibly requiring a large number of iterations. Consequently the total number of iterations for convergence may become excessively large for practical applications.

In order to accelerate convergence, we have adopted the modifications on the updating rules proposed by Miele *et al.* (1972) for the method of the multipliers. The adopted modifications are the following:

(i) Shorten the length of a cycle of computations. A cycle of computations is defined to be the sequence of iterations in which the multiplier λ and the penalty constant K_m are held unchanged, while the vector q_j for the subsystem j is viewed as unconstrained. The number of iterations permitted in each cycle depends on the unconstrained optimization technique which is employed, and it is ΔN = number of iterations = 1 for the ordinary gradient algorithm and the modified quasi-linearization algorithm (Miele *et al.*, 1972) and ΔN = dimension of vector q_j , for the conjugate-gradient algorithm.

(ii) Improve the estimate of the Lagrange multipliers λ , using the formula

$$\lambda^{(i+1)} = \lambda^{(i)} + 2\beta g(q)$$

where β is a scalar parameter determined so as to produce some optimum effect, namely, to minimize the error in the necessary conditions for optimality.

(iii) Select the penalty constant K_m in an appropriate fashion.

The method used to select K_m depends largely on the unconstrained optimization method which is employed. If the ordinary gradient method is employed, the penalty constant is given by the formula

$$K_m = 2P(q)/P_q^T(q) P_q(q) ,$$

where $P(q) = g^T(q)g(q)$ and $P_q(q) = \frac{\partial P(q)}{\partial q}$. If the conjugate-gradient or the modified quasi-linearization algorithm is used, the penalty constant is updated by the formula

$$K_m^{(i+1)} = \min (K_0, K_m^{(i)}) \quad \text{if} \quad P(q) \leq Q(q, \lambda)$$

$$K_m^{(i+1)} = \max (K_0, \pi K_m^{(i)}) \quad \text{if} \quad P(q) > Q(q, \lambda)$$

where: $Q(q, \lambda) = F_q^T(q, \lambda) F_q(q, \lambda) , \quad F_q(q, \lambda) = \frac{\partial F}{\partial q} - \frac{\partial g^T}{\partial q} \lambda, n \geq 1$

and

$$K_0 = \left[\begin{bmatrix} \frac{\partial g(q)}{\partial q} \lambda \\ \vdots \\ \frac{\partial g(q)}{\partial q} \lambda \end{bmatrix}^T \begin{bmatrix} \frac{\partial g(q)}{\partial q} \lambda \\ \vdots \\ \frac{\partial g(q)}{\partial q} \lambda \end{bmatrix} / P_q^T(q) P_q(q) \right] .$$

See Miele et al. (1972) for further information on these rules.

II.7 A Discrete Minimum Principle

In this section we shall give a theoretical justification for the algorithm given in Section II.6. Consider the following two Lagrangian functions:

$$L_m^* = F(q) + K_m g^T(q)g(q) - \lambda^T g(q) \quad (\text{II-10})$$

and

$$\hat{L}_m^* = F(q) + K_m(g(q) - z)^T (g(q) - z) + \mu^T (g(q) - z). \quad (\text{II-11})$$

Geometrically (II-11) is the Lagrange function resulting from moving the origin to z in the WZ space and adding the penalty term for deviation from that new origin. Theorem 8 indicates a relationship which exists between them.

Theorem 8: If \tilde{q} solves the problem

$$h_m^*(\lambda) = \min_{q \in S} L_m^*(q, K_m, \lambda) \quad (\text{II-P3})$$

resulting in

$$g(\tilde{q}) = z \quad (\text{II-12})$$

then \tilde{q} also solves, for K_m and z fixed at these same values, the problem

$$\min_{q \in S} \hat{L}_m^*(q, K_m, \mu, z) | g(q) = z \}. \quad (\text{II-P4})$$

Conversely, if one can find a z which permits (II-P4) to have a solution, say \tilde{q} , then \tilde{q} solves a problem of form (II-P3).

Proof: Suppose that \tilde{q} solves (II-P3) and results in $g(\tilde{q}) = z$. Then by Everett's main theorem (Everett, 1963), q solves the problem

$$\min_{q \in S} \{F(q) + K_m^T(q)g(q) | g(q) = z\} \quad (\text{II-13})$$

which is equivalent to solving the problem

$$\min_{q \in S} \{F(q) | g(q) = z\} \quad (\text{II-14})$$

since the term $K_m^T q g(q) = K_m^T z z$ is constant for z fixed. Problem (II-P4) is equivalent to problem (II-14); thus \tilde{q} solves problem (II-P4).

If z is fixed and permits (II-P4) to have a solution, say \tilde{q} , then one has solved the problem

$$\min_{q \in S} \{F(q) + K_m^T q g(q) - (2K_m z + \mu)^T g(q) + K_m^T z z + \mu^T z\} ,$$

which is equivalent to solving the problem

$$\min_{q \in S} \{F(q) + K_m^T q g(q) - \lambda^T g(q)\} \quad (\text{II-15})$$

where

$$\lambda = 2K_m z + \mu . \quad (\text{II-16})$$

Problem (II-15) is a problem of form (II-P3), and consequently \tilde{q} solves a problem of form (II-P3).

Theorem 8 says problems (II-P3) and (II-P4) are equivalent, in that they give rise to the same (z, \tilde{q}) values. The following corollary follows directly from this observation.

Corollary 8.1: If and only if \hat{z} permits (II-P4) to have a solution, then the set R_m corresponding to problem (II-P3) has a supporting hyperplane at the point $(w_m^*(\hat{z}), \hat{z})$ with a "slope" given by (II-16).

The following corollary also follows directly.

Corollary 8.2: If \tilde{z} permits (II-P4) to have a solution, say \tilde{q} , then

$$h(\lambda) = \mathcal{L}_m^*(\tilde{q}, K_m, \mu; \tilde{z}) + K_m \tilde{z}^T \tilde{z} + \lambda^T \tilde{z} \quad (\text{II-17})$$

where λ is given by (II-16).

Proof: If z permits (II-P4) to have a solution, \tilde{q} , then

$$g(\tilde{q}) = \tilde{z} \quad (\text{II-18})$$

and

$$\mathcal{L}_m^*(\tilde{q}, K_m, \mu, \tilde{z}) = F(\tilde{q}) . \quad (\text{II-19})$$

Equation (II-17) follows immediately from (II-10), (II-18) and

(II-19). Q.E.D.

Note that corollary 8.2 gives us a formula to calculate a dual bound, $h(\lambda)$, to $w(0)$ if we have in fact solved (II-P4) rather than (II-P3).

The result following exposes some very useful properties of problem (II-P4) for the type of engineering design problems being considered in this paper. It also constitutes the theoretical reasoning of the algorithm presented in Section II-6.

Result: If the point $\tilde{q}^T = [\tilde{q}_1^T, \dots, \tilde{q}_n^T]$ and the multipliers μ solve problem (II-P4), then each subproblem j , $j=1, \dots, n$ resulting after the Taylor Series linear approximation of the crossproduct terms is minimized with respect to the corresponding q_j at the point \tilde{q}_j .

Proof: Consider a system of two stages and the following family of problems:

$$\text{Min } F = f_1(x_1, u_1) + f_2(x_2, u_2)$$

$$\text{subject to: } x_2 - t_1(x_1, u_1) = z$$

$$h_1(x_1, u_1) \leq 0$$

$$h_2(x_2, u_2) \leq 0$$

where $z \in Z' = \{z | x_2 - t_1(x_1, u_1) = z, h_1(x_1, u_1) \leq 0, \text{ and } h_2(x_2, u_2) \leq 0\}$.

The Lagrangian function for the augmented problem is:

$$L_m^* = f_1(x_1, u_1) + f_2(x_2, u_2) + \kappa_m [x_2 - t_1(x_1, u_1) - z]^T [x_2 - t_1(x_1, u_1) - z]$$

$$+ v_1^T [x_2 - t_1(x_1, u_1) - z] + v_1^T h_1(x_1, u_1) + v_2^T h_2(x_2, u_2)$$

$$= L + \kappa_m [x_2 - t_1(x_1, u_1) - z]^T [x_2 - t_1(x_1, u_1) - z] ,$$

where v_1 and v_2 are Kuhn-Tucker multipliers.

Suppose that L_m^* is minimized at the point $(\tilde{x}_1, \tilde{u}_1, \tilde{x}_2, \tilde{u}_2)$. At this point

the Hessian matrix of the second derivatives of L_m^* must be positive definite, i.e.,

$$A =$$

$$\begin{bmatrix} \frac{\partial^2 L}{\partial x_1^2} + 2K_m \frac{\partial t_1^T}{\partial x_1} \frac{\partial t_1}{\partial x_1} - 2K_m \frac{\partial^2 t_1}{\partial x_1^2} [x_2 - t_1 - z] & \frac{\partial^2 L}{\partial x_1 \partial u_1} + 2K_m \frac{\partial t_1^T}{\partial x_1} \frac{\partial t_1}{\partial u_1} - 2K_m \frac{\partial^2 t_1}{\partial x_1 \partial u_1^T} [x_2 - t_1 - z] & \frac{\partial^2 L}{\partial x_1 \partial u_1^T} + 2K_m \frac{\partial t_1^T}{\partial x_1^T} \frac{\partial t_1}{\partial u_1^T} - 2K_m \frac{\partial^2 t_1}{\partial x_1^T \partial u_1^T} [x_2 - t_1 - z] \\ \frac{\partial^2 L}{\partial x_1^T \partial u_1} + 2K_m \frac{\partial t_1^T}{\partial u_1} \frac{\partial t_1}{\partial x_1} - 2K_m \frac{\partial^2 t_1}{\partial u_1 \partial x_1^T} [x_2 - t_1 - z] & \frac{\partial^2 L}{\partial u_1^2} + 2K_m \frac{\partial t_1^T}{\partial u_1} \frac{\partial t_1}{\partial u_1^T} - 2K_m \frac{\partial^2 t_1}{\partial u_1^2} [x_2 - t_1 - z] & \frac{\partial^2 L}{\partial u_1^T \partial u_1} + 2K_m \frac{\partial t_1^T}{\partial u_1^T} \frac{\partial t_1}{\partial u_1} - 2K_m \frac{\partial^2 t_1}{\partial u_1^T \partial u_1} [x_2 - t_1 - z] \\ - 2K_m \frac{\partial t_1}{\partial x_1^T} & - 2K_m \frac{\partial t_1}{\partial u_1^T} & 0 \end{bmatrix}$$

must be positive definite.

Note that A contains terms which are not separable, each of which contains the term $x_2 - t_1 - z$. Since \tilde{u} and $(x_1, \tilde{u}_1, \tilde{x}_2, \tilde{u}_2)$ minimize \mathcal{L}_m^* with $\tilde{x}_2 - t_1(\tilde{x}_1, \tilde{u}_1) - z = 0$, the matrix A reduces to the following form:

$$A = \begin{bmatrix} \frac{\partial^2 L}{\partial x_1^2} + 2K_m \frac{\partial t_1^T}{\partial x_1} \frac{\partial t_1}{\partial x_1^T} & \frac{\partial^2 L}{\partial x_1 \partial u_1^T} + 2K_m \frac{\partial t_1^T}{\partial x_1} \frac{\partial t_1}{\partial u_1^T} & - 2K_m \frac{\partial t_1^T}{\partial x_1} & 0 \\ \frac{\partial^2 L}{\partial x_1^T \partial u_1} + 2K_m \frac{\partial t_1^T}{\partial u_1} \frac{\partial t_1}{\partial x_1^T} & \frac{\partial^2 L}{\partial u_1^2} + 2K_m \frac{\partial t_1^T}{\partial u_1} \frac{\partial t_1}{\partial u_1^T} & - 2K_m \frac{\partial t_1^T}{\partial u_1} & 0 \\ - 2K_m \frac{\partial t_1}{\partial x_1^T} & - 2K_m \frac{\partial t_1}{\partial u_1^T} & \frac{\partial^2 L}{\partial x_2^2} + 2K_m I & \frac{\partial^2 L}{\partial x_2^T \partial u_2} \\ 0 & 0 & \frac{\partial^2 L}{\partial x_2^T \partial u_2} & \frac{\partial^2 L}{\partial u_2^2} \end{bmatrix}$$

Since A must be positive definite the principal minor matrices of A must be positive definite, i.e.

$$A_{ij} \Big|_{\tilde{x}_1, \tilde{u}_1} = \begin{bmatrix} \frac{\partial^2 L}{\partial x_1^2} + 2K_m \frac{\partial t_1^T}{\partial x_1} \frac{\partial t_1}{\partial x_1^T} & \frac{\partial^2 L}{\partial x_1 \partial u_1^T} + 2K_m \frac{\partial t_1^T}{\partial x_1} \frac{\partial t_1}{\partial u_1^T} \\ \frac{\partial^2 L}{\partial x_1^T \partial u_1} + 2K_m \frac{\partial t_1^T}{\partial u_1} \frac{\partial t_1}{\partial x_1^T} & \frac{\partial^2 L}{\partial u_1^2} + 2K_m \frac{\partial t_1^T}{\partial u_1} \frac{\partial t_1}{\partial u_1^T} \end{bmatrix} \quad \text{positive definite}$$

and

$$A_2 \begin{bmatrix} \tilde{x}_2 \\ \tilde{u}_2 \end{bmatrix} = \begin{bmatrix} \frac{\partial^2 L}{\partial x_2^2} + 2K_m I & \frac{\partial^2 L}{\partial x_2 \partial u_2^T} \\ \frac{\partial^2 L}{\partial x_2^T \partial u_2} & \frac{\partial^2 L}{\partial u_2^2} \end{bmatrix} \text{ positive definite .}$$

Consider now the sub-Lagrangians ℓ_{m1}^* and ℓ_{m2}^* . After the linear approximation of the crossproduct terms $-2K_m x_2^T t_1(x_1, u_1)$, they take the following form:

$$\begin{aligned} \ell_{m1}^* = f_1(x_1, u_1) - \tilde{u}^T t_1(x_1, u_1) - v_1^T h_1(x_1, u_1) + K_m t_1^T(x_1, u_1) t_1(x_1, u_1) \\ - 2K_m \hat{x}_2^T t_1(x_1, u_1) - 2K_m z^T t_1 \end{aligned}$$

$$\ell_{m2}^* = f_2(x_2, u_2) + \tilde{u}^T x_2 - v_2^T h_2(x_2, u_2) + K_m x_2^T x_2 - 2K_m t_1^T(\hat{x}_1, \hat{u}_1) x_2 + 2K_m z^T x_2$$

where $\hat{x}_1, \hat{u}_1, \hat{x}_2$ is the point around which the Taylor Series expansion takes place. At the point $\tilde{x}_1, \tilde{u}_1, \tilde{x}_2, \tilde{u}_2$ the following conditions are satisfied:

$$\frac{\partial \ell_{m1}^*}{\partial x_1} = \frac{\partial \ell_{m1}^*}{\partial x_1} = 0 , \quad \frac{\partial \ell_{m1}^*}{\partial u_1} = \frac{\partial \ell_{m1}^*}{\partial u_1} = 0$$

$$\frac{\partial \ell_{m2}^*}{\partial x_2} = \frac{\partial \ell_{m2}^*}{\partial x_2} = 0 \quad \text{and} \quad \frac{\partial \ell_{m2}^*}{\partial u_2} = \frac{\partial \ell_{m2}^*}{\partial u_2} = 0 .$$

Also the Hessian matrices of ℓ_{m1}^* and ℓ_{m2}^* , which are precisely A_1 and A_2 respectively, are positive definite. Thus we conclude that ℓ_{m1}^* and ℓ_{m2}^*

are minimized at the point $(\tilde{x}_1, \tilde{u}_1, \tilde{x}_2, \tilde{u}_2)$ which minimizes L_m^* . By induction the same result is found to apply to any number of stages.

Since the primal problem (II-P1) corresponds to that member of the class of problems (II-P1') with $z = 0$, the above mentioned result applies to (II-P1) for a proper K_m and μ .

II.8. Discussion

We should note the following two observations which constitute the essence of the proposed algorithm: 1) The use of a Taylor Series expansion really provided two results. The first is, of course, that the problem becomes separable. The second is evident only when one realizes that other devices could effect the separability feature; for example, one could rewrite the crossproduct term $-2x_{ir}t_{ir}$ as $-2x_{ir}(x_{ir} - z_{ir})$ which is also separable for fixed z_{ir} . Unfortunately, in this form in contrast to the proposed one, the subproblem, for the unit containing x_{ir} as a variable, becomes a concave minimization in x_{ir} as K_m increases. 2) For the Taylor Series expansion approach, we have found that problem (II-P4), and not (II-P3) to which it is equivalent, has the desired property that the subproblems must always be minimized if the overall problem is minimized, even if K_m is not sufficiently large.

We should also note that the convex quadratic terms t_{ir}^2 and x_{ir}^2 produced by the penalty term will tend to improve and could dominate the behavior of the subproblems as K_m increases, making them easier to solve.

The introduction of the quadratic terms in the objective function also offers another advantage. It desensitizes the dual function with

respect to the multipliers λ . For given multipliers $\tilde{\lambda}$,

$$h^*(\tilde{\lambda}) = \min_{x_i, u_i, i=1, \dots, n} L^* \quad \text{is closer to the primal solution than the}$$

$$h(\tilde{\lambda}) = \min_{x_i, u_i, i=1, \dots, n} L \quad (\text{see Figure 3}). \quad \text{This is a characteristic}$$

which can be of importance for a dual bounding procedure, such as the one used in structural sensitivity analysis (McGalliard, Westerberg, 1972).

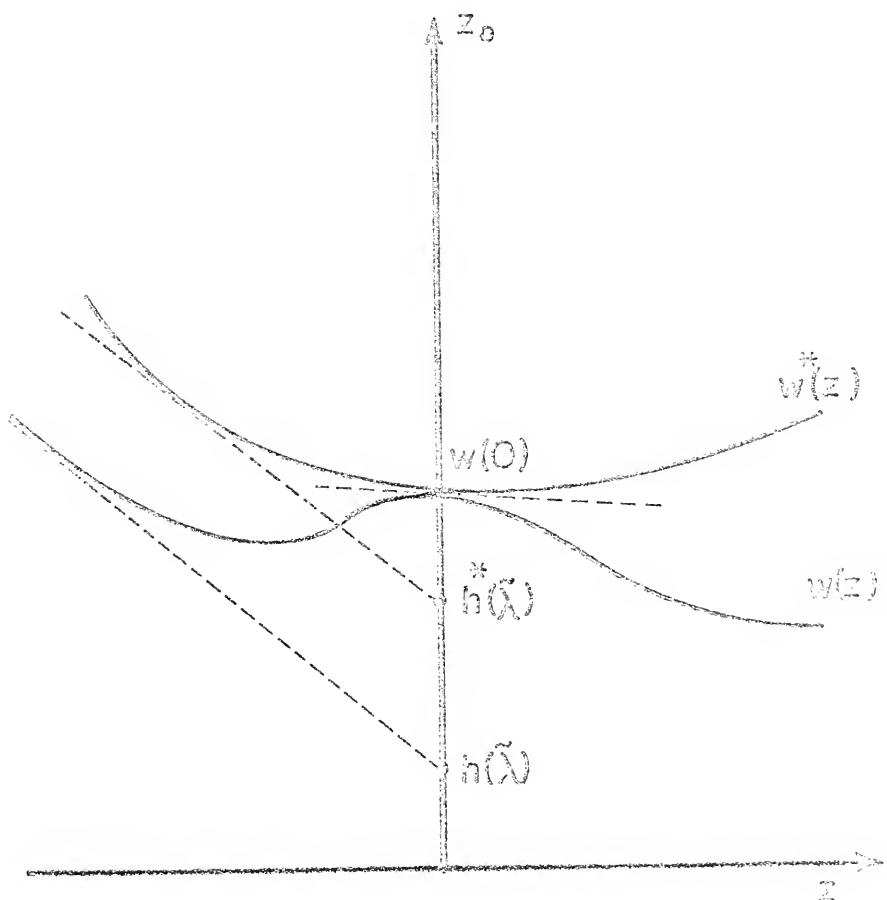


Figure 3. Improvement of the Dual Bound $h^*(\tilde{\lambda})$ for the Penalized Problem Over the Dual Bound $h(\tilde{\lambda})$ for the Unpenalized Problem for the Same Multipliers $\tilde{\lambda}$

CHAPTER III

A STRONG VERSION OF THE DISCRETE MINIMUM PRINCIPLE

The discrete form of Pontryagin's Minimum Principle proposed by a number of authors has been shown by others in the past to be fallacious; only a weak result can be obtained. Due to the mathematical character of the objective function and the stage transformation equations, only a small class of chemical engineering problems have been solved by the strong discrete minimum principle. This chapter presents a method to overcome the previous shortcomings of the strong principle. An algorithmic procedure is developed which uses this new version. Numerical examples are provided to clarify the approach and demonstrate its usefulness.

III.1. Review of Previous Works

Pontryagin's minimum principle (Pontryagin *et al.*, 1962) is a well-known method to solve a wide class of extremal problems associated with given initial conditions. A discrete analog of the minimum principle, where the differential equations are substituted by difference equations, is not valid in general but only in certain almost trivial cases. Rozonoer (1959) first pointed out this fact. Katz (1962) and Fan and Wang (1964) later on developed a discrete minimum principle which was shown to be fallacious by Horn and Jackson (1965a), by means of simple counterexamples. As was pointed out by Horn and Jackson

(1965a, b) and lucidly presented by Denn (1969), the failure of a strong minimum principle lies in the fact that we cannot deduce the nature of the stationary values of the Hamiltonian from a consideration of first-order variations only. Inclusion of the second-order terms, does not help to draw a general conclusion about the nature of the stationary points in advance. A weak minimum principle which relates the solution of the problem to a stationary point of the Hamiltonian exists and is valid (Horn, 1961 ;Jackson, 1964).

In the case of control systems described by differential equations, time, by its evolution on a continuum, has a "convexifying" effect (Halkin, 1966) which does not make necessary the addition of some convexity assumptions to the specification of the problem. Thus a strong minimum principle can be applied for these problems, requiring the minimization of the Hamiltonian even in the case that a continuous problem is solved by discretizing it with respect to the time and using a strong discrete minimum principle. For discrete, staged systems described by difference equations, the evolution of the system does not have any "convexifying" effect and, in order to obtain a minimum principle, we must add some convexity assumptions to the problem specification or reformulate the problem in an equivalent form which possesses inherently the convexity assumptions. This present work belongs to the second class.

In the present chapter we propose to show a strong version of the minimum principle which relates the solution of the problem to a minimum point, rather than a stationary point, of the Hamiltonian. This is attained through the use of the Hestenes' method of multipliers, a technique used effectively in Chapter II to resolve the dual gaps

of the two-level optimization method. This method turns a stationary point of the Hamiltonian into a minimum point, thus minimum seeking algorithms can be used.

III.2. Statement of the Problem, and Its Lagrangian Formulation

As a basis for the description of the minimum principle, the two-level optimization approach, their success and failure, and the development of a strong minimum principle, consider the following sequential unconstrained problem. (Constraints and recycles do not change the following results (Westerberg, 1973), and we want to keep the presentation here as simple as possible.)

$$\text{Min } F = \sum_{i=1}^N \phi_i(x_i, u_i) \quad (\text{III-P1})$$

subject to

$$x_{i+1} = f_i(x_i, u_i)$$

$$x_1 = x_1^0 \text{ (given)} .$$

For every $i=2, \dots, N$ the vector valued function $f_i(x_i, u_i)$ is given and satisfies the following conditions:

- the function f_i is defined for all x_i and u_i ,
- for every u_i the function $f_i(x_i, u_i)$ is twice continuously differentiable with respect to x_i ,
- the $f_i(x_i, u_i)$ and all its first and second partial derivatives are uniformly bounded.

These conditions correspond to the usual "smoothness" assumptions.

The Lagrangian function for this problem (III-P1) is given by:

$$\begin{aligned}
 L &= \sum_{i=1}^N \phi_i(x_i, u_i) - \sum_{i=1}^N \lambda_{i+1}^T [x_{i+1} - f_i(x_i, u_i)] - \lambda_1^T (x_1 - x_1^0) \\
 &= \sum_{i=1}^N \{\phi_i(x_i, u_i) - \lambda_i^T x_i + \lambda_{i+1}^T f_i(x_i, u_i)\} + \lambda_1^T x_1^0 - \lambda_{N+1}^T x_{N+1} \\
 &= \sum_{i=1}^N \phi_i(x_i, u_i, \lambda_i, \lambda_{i+1}) + \lambda_1^T x_1^0 - \lambda_{N+1}^T x_{N+1} . \tag{III-1}
 \end{aligned}$$

The solution to the problem (III-P1) is a stationary point of the Lagrangian function L . The necessary conditions for a stationary point of the Lagrangian are:

$$\frac{\partial L}{\partial x_i} = \frac{\partial \phi_i}{\partial x_i} - \lambda_i + \frac{\partial f_i^T}{\partial x_i} \lambda_{i+1} = 0, \quad i=1, \dots, N \tag{III-2}$$

$$\frac{\partial L}{\partial u_i} = \frac{\partial \phi_i}{\partial u_i} + \frac{\partial f_i^T}{\partial u_i} \lambda_{i+1} = 0, \quad i=1, \dots, N \tag{III-3}$$

$$\frac{\partial L}{\partial \lambda_i} = x_i - f_{i-1}(x_i, u_i) = 0, \quad i=1, \dots, N . \tag{III-4}$$

From equation (III-2) we have the defining equations for the multipliers,

$$\lambda_i = \frac{\partial \phi_i}{\partial x_i} + \frac{\partial f_i^T}{\partial x_i} \lambda_{i+1} \tag{III-5}$$

with the natural boundary condition,

$$\lambda_{N+1} = 0 . \tag{III-6}$$

Equation (III-4) simply necessitates the satisfaction of the connection constraints.

The Lagrangian approach constitutes a unifying and general presentation of the necessary conditions which must be satisfied at the solution of a problem. For the solution of the necessary conditions different strategies have been developed. In a tutorial presentation (Westerberg, 1973) the relationship of the different strategies to solve problem (III-P1), with the Lagrangian approach, is established and it is shown that methods such as sensitivity analysis, discrete minimum principle, and the two-level optimization method are simply different techniques to solve the same necessary conditions, eqs. (III-1), (III-2) and (III-3).

Let us define the stage Hamiltonian H_i as follows:

$$H_i = \phi_i(x_i, u_i) + \lambda_{i+1}^T f_i(x_i, u_i) \quad i=1, \dots, N \quad (\text{III-7})$$

and the overall Hamiltonian by:

$$H = \sum_{i=1}^N H_i + \lambda_1^T x_1^0 .$$

Then,

$$L = \sum_{i=1}^N H_i - \sum_{i=1}^N \lambda_i x_i + \lambda_1^T x_1^0$$

and the necessary conditions (III-2), (III-3) yield:

$$\frac{\partial L}{\partial x_i} = \frac{\partial H_i}{\partial x_i} - \lambda_i = 0 , \quad i=1, \dots, N \quad (\text{III-8})$$

$$\frac{\partial L}{\partial u_i} = \frac{\partial H_i}{\partial u_i} = 0 , \quad i=1, \dots, N . \quad (\text{III-9})$$

Thus, in order to solve problem (III-P1) by the discrete minimum principle, we require that each stage Hamiltonian be at a stationary point.

Consider the point $(\tilde{x}_1, \tilde{x}_2, \dots, \tilde{x}_N; \tilde{u}_1, \tilde{u}_2, \dots, \tilde{u}_N)$ and the variations $\delta u_1, \delta u_2, \dots, \delta u_N$ around the previous point with respect to the controls, such that $|\delta u_{ij}| \leq \varepsilon$, where the second index j denotes the j th element of the control vector u_i . δx_1 will be taken as zero. The variational equations corresponding to the connection constraints of the problem (III-P1), with up to the second order terms included, are:

$$\begin{aligned} \delta x_{i+1} = & \frac{\partial f_i}{\partial u_i^T} (\delta u_i) + \frac{\partial f_i}{\partial x_i^T} (\delta x_i) + \frac{1}{2} (\delta u_i)^T \frac{\partial^2 f_i}{\partial u_i^2} (\delta u_i) + \frac{1}{2} (\delta x_i)^T \frac{\partial^2 f_i}{\partial x_i^2} (\delta x_i) \\ & + (\delta u_i)^T \frac{\partial^2 f_i}{\partial u_i \partial x_i^T} (\delta x_i) + O(\varepsilon^2) . \end{aligned}$$

For the considered sequential system, the solution of the above system with respect to the δx_i 's, $i=1, \dots, N$, is straightforward and yields the following general formula in terms of the variations in the controls only:

$$\begin{aligned} \delta x_{i+1} = & \sum_{\ell=1}^i \left[\begin{matrix} i & \frac{\partial f_k^T}{\partial x_k} \\ \Pi & \frac{\partial f_k^T}{\partial x_k} \end{matrix} \right]^T \frac{\partial f_\ell}{\partial u_\ell^T} (\delta u_\ell) + \frac{1}{2} \sum_{\ell=1}^i \left[\begin{matrix} i & \frac{\partial f_k^T}{\partial x_k} \\ \Pi & \frac{\partial f_k^T}{\partial x_k} \end{matrix} \right]^T (\delta u_\ell)^T \frac{\partial^2 f_\ell}{\partial u_\ell^2} (\delta u_\ell) \\ & + \frac{1}{2} \sum_{\ell=1}^i \sum_{k,m=1}^\ell \left[\begin{matrix} i & \frac{\partial f_v^T}{\partial x_v} \\ \Pi & \frac{\partial f_v^T}{\partial x_v} \end{matrix} \right]^T (\delta u_k)^T \frac{\partial f_k^T}{\partial u_k} \left[\begin{matrix} \ell-1 & \frac{\partial f_s^T}{\partial x_s} \\ \Pi & \frac{\partial f_s^T}{\partial x_s} \end{matrix} \right] \frac{\partial^2 f_\ell}{\partial x_\ell^2} \left[\begin{matrix} \ell-1 & \frac{\partial f_t^T}{\partial x_t} \\ \Pi & \frac{\partial f_t^T}{\partial x_t} \end{matrix} \right]^T \\ & \frac{\partial f_m^T}{\partial u_m^T} (\delta u_m) + \frac{1}{2} \sum_{\ell=1}^i \sum_{k=1}^\ell \left[\begin{matrix} i & \frac{\partial f_v^T}{\partial x_v} \\ \Pi & \frac{\partial f_v^T}{\partial x_v} \end{matrix} \right]^T (\delta u_k)^T \frac{\partial f_k^T}{\partial u_k} \left[\begin{matrix} \ell-1 & \frac{\partial f_s^T}{\partial x_s} \\ \Pi & \frac{\partial f_s^T}{\partial x_s} \end{matrix} \right] \frac{\partial^2 f_\ell}{\partial x_\ell \partial u_\ell^T} (\delta u_\ell) . \end{aligned}$$

The variation in the objective function F caused by the variations in the controls is given by:

$$\delta F = \sum_{i=1}^N \left[\frac{\partial \phi_i}{\partial u_i^T} (\delta u_i) + \frac{\partial \phi_i}{\partial x_i} (\delta x_i) + \frac{1}{2} (\delta u_i)^T \frac{\partial^2 \phi_i}{\partial u_i^2} (\delta u_i) \right. \\ \left. + \frac{1}{2} (\delta x_i)^T \frac{\partial^2 \phi_i}{\partial x_i^2} (\delta x_i) + (\delta u_i)^T \frac{\partial^2 \phi_i}{\partial u_i \partial x_i} (\delta x_i) \right] + O(\varepsilon^2).$$

Substituting into the last expression for δx_i 's, $i=2, \dots, N$ with their equals from eqs. (III-10) and, noting that the stage Hamiltonians H_i , $i=1, \dots, N$, are given by eq. (III-7), we find:

$$\delta F = \sum_{\ell=1}^N \left[\frac{\partial H_\ell}{\partial u_\ell^T} (\delta u_\ell) + \frac{1}{2} (\delta u_\ell)^T \frac{\partial^2 H_\ell}{\partial u_\ell^2} (\delta u_\ell) \right] \\ + \frac{1}{2} \sum_{\ell=1}^N \sum_{k=1}^{\ell} (\delta u_k)^T \frac{\partial f_k^T}{\partial u_k} \left[\begin{array}{cc} \ell-1 & \frac{\partial f_s^T}{\partial x_s} \\ \vdots & \frac{\partial f_s^T}{\partial x_s} \\ s=k+1 & \end{array} \right] \frac{\partial^2 H_\ell}{\partial x_\ell^2} \left[\begin{array}{cc} \ell-1 & \frac{\partial f_t^T}{\partial x_t} \\ \vdots & \frac{\partial f_t^T}{\partial x_t} \\ t=m+1 & \end{array} \right]^T \frac{\partial f_m}{\partial u_m^T} (\delta u_m) \\ + \frac{1}{2} \sum_{\ell=1}^N \sum_{k=1}^N (\delta u_k)^T \frac{\partial f_k^T}{\partial u_k} \left[\begin{array}{cc} \ell-1 & \frac{\partial f_s^T}{\partial x_s} \\ \vdots & \frac{\partial f_s^T}{\partial x_s} \\ s=k+1 & \end{array} \right] \frac{\partial^2 H_\ell}{\partial x_\ell \partial u_\ell^T} (\delta u_\ell). \quad (\text{III-11})$$

Unlike the continuous case (Denn, 1969; Halkin, 1966) there is no general way in which the last two terms in eq. (III-11) can be made to vanish. Therefore, the variations considered in the controls may well produce $\delta F < 0$, or $\delta F > 0$, or $\delta F = 0$. Thus it is evident from the above that a strong minimum principle, which requires that the solution to the problem (III-P1) minimizes the stage Hamiltonians H_ℓ , $\ell=1, \dots, N$, is not generally available (Horn and Jackson, 1965b; Denn, 1969; Halkin, 1966). A weaker form of the discrete minimum principle can

be used and requires that the solution to the problem (III-P1) makes the stage Hamiltonians stationary. The examples presented by Horn and Jackson, which counter the strong minimum principle, are such that the stage Hamiltonians do not possess a minimum stationary point whereas the problem itself does. However, there exist special cases where the strong minimum principle can be applied (Horn and Jackson, 1965b; Denn, 1969).

At this point a further clarification is required. The strong discrete minimum principle fails for physically staged steady-state systems. Halkin (1966) has shown that it always succeeds for discrete-time systems, obtained as an approximation to the continuous time systems, since the time increment t can become arbitrarily small and make the last two terms in eq. (III-11) vanish. The success or the failure of the strong discrete minimum principle can also be explained in terms of convexity (or directional convexity) or lack of it for the sets of the reachable states of the system (Halkin, 1966; Holtzman and Halkin, 1966). This constitutes an important characteristic and in fact it is the basis for the development of a strong version of the discrete minimum principle, which follows.

The two-level optimization procedure is an infeasible decomposition strategy of a Lagrangian nature which solves the problem (III-P1). The similarity between the Lagrangian and the weak discrete minimum principle is a well-known fact and in a recent note (Schock and Luus, 1972) the similarity between the two-level optimization procedure and the discrete minimum principle has been pointed out. Further insight in the relationship between the last two methods will be given later in this chapter, and a stronger version of the discrete minimum

principle will be developed, based on a method to overcome the shortcomings of the two-level optimization procedure developed in Chapter II.

III.3. Development of a Stronger Version of the Discrete Minimum Principle

In this section we will develop a stronger version of the discrete minimum principle using Hestenes' method of multipliers and following a procedure similar to the one that was developed in Chapter II to overcome the dual gaps of the two-level optimization procedure.

As a basis for the presentation we will use problem (III-P1). Consider now the augmented problem (III-P2)

$$\text{Min } F^* = \sum_{i=1}^N \dot{\phi}_i(x_i, u_i) + K \sum_{i=1}^N [x_{i+1} - f_i]^T [x_{i+1} - f_i] + K(x_1 - x_1^0)^T (x_1 - x_1^0)$$

subject to (III-P2)

$$x_{i+1} = f_i(x_i, u_i) \quad i=1, \dots, N$$

$$x_1 = x_1^0 \quad (\text{given})$$

with $K > 0$.

The Hamiltonian for this problem is given by:

$$H^* = F^* + \sum_{i=1}^N \lambda_{i+1}^T f_i(x_i, u_i) + \lambda_1^T x_1^0 .$$

In the following theorems we will establish some important properties of H^* .

Theorem 9: If $(\tilde{u}_1, \dots, \tilde{u}_N)$ is a stationary point of the Hamiltonian H for the problem (III-P1), then it is also a stationary point for the

Hamiltonian H^* of the augmented problem (III-P2) for any real value of the parameter K .

Proof: From the equations defining H and H^* we take,

$$H^* = H + K \sum_{i=1}^N [x_{i+1} - f_i]^T [x_{i+1} - f_i] + K(x_1 - x_1^0)^T (x_1 - x_1^0). \quad (\text{III-12})$$

Differentiation with respect to the controls yields:

$$\frac{\partial H^*}{\partial u} = \frac{\partial H}{\partial u} + 2K \sum_{i=1}^N \frac{\partial [x_{i+1} - f_i]^T}{\partial u} [x_{i+1} - f_i].$$

Evaluating the derivatives at the point $(\tilde{u}_1, \dots, \tilde{u}_N)$, since $\frac{\partial H}{\partial u}\Big|_{\tilde{u}} = 0$

and $x_{i+1} - f_i = 0$ for $i=1, \dots, N$, and we find

$$\frac{\partial H^*}{\partial u} = \frac{\partial H}{\partial u} = 0.$$

Theorem 10: Let $u=\tilde{u}$ be a local isolated minimum of F with all the connection constraints satisfied. Then there exists a real valued parameter K_0 , such that $\frac{\partial H^*}{\partial u}\Big|_{\tilde{u}} = 0$ and $\frac{\partial^2 H^*}{\partial u^2}\Big|_{\tilde{u}} > 0$ (i.e., the matrix

is positive definite) for $K > K_0$.

Proof: In Theorem 9 it was shown that $\frac{\partial H^*}{\partial u}\Big|_{\tilde{u}} = 0$ is independent of the

constant K ; therefore, it will be valid and for $K > K_0$. Let us consider next the matrix $\frac{\partial^2 H^*}{\partial u^2}$:

$$\frac{\partial^2 H^*}{\partial u^2} = \frac{\partial^2 H}{\partial u^2} + 2K \sum_{i=1}^N \frac{\partial [x_{i+1} - f_i]^T}{\partial u} \frac{\partial [x_{i+1} - f_i]}{\partial u}.$$

The solution of problem (III-P1) is a stationary point of the Hamiltonian H . The nature of the stationary point for the discrete case cannot be predetermined and thus $\frac{\partial^2 H}{\partial u^2} \leq 0$. Assuming that we have a nonsingular problem, i.e.

$$\frac{\partial [x_{i+1} - f_k]^T}{\partial u} \quad \frac{\partial [x_{i+1} - f_i]^T}{\partial u} \neq 0 \quad i=1, \dots, N$$

and we conclude that

$$(\delta u)^T \frac{\partial^2 H^*}{\partial u^2} (\delta u) = (\delta u)^T \frac{\partial^2 H}{\partial u^2} (\delta u) + 2K \sum_{i=1}^N \left[\frac{\partial [x_{i+1} - f_i]^T}{\partial u} (\delta u) \right]^T .$$

$$\left[\frac{\partial [x_{i+1} - f_i]^T}{\partial u} (\delta u) \right] > 0$$

for a large enough K so that the second term which is always positive prevails over any negativity of the first term. Q.E.D.

The second theorem implies that any stationary point of the Hamiltonian H can be made a minimum point by choosing the parameter K large enough.

Theorems 9 and 10 establish the following result: For a large enough K , a local solution to problem (III-P2) which is also a local solution to problem (III-P1), minimizes the Hamiltonian H^* of the augmented problem (III-P2).

The above result requires that we minimize the overall problem Hamiltonian, H^* . The available weak minimum principle permits us to solve the overall problem by solving one problem per stage per

iteration since H decomposes into a sum of stage Hamiltonians.

Unfortunately H^* does not decompose, as we shall now see.

Let us see now how we can simplify the above result and relate the solution of the problem (III-P1) or (III-P2) to stage Hamiltonians. Consider the penalty term in the objective function of the problem (III-P2):

$$K = \sum_{i=1}^N [x_{i+1} - f_i]^T [x_{i+1} - f_i] = K \sum_{i=1}^N [x_{i+1}^T x_{i+1} + f_i^T f_i - 2x_{i+1}^T f_i]$$

We note that each member of the summation includes separable terms, e.g., $x_{i+1}^T x_{i+1}$ and $f_i^T f_i$, and nonseparable terms such as the cross-product $x_{i+1}^T f_i$. The following develops a strategy to decompose computationally the problem of minimizing H^* . This approach is motivated by the work on solving nonconvex problems using the two-level approach.

Expand the crossproduct term in a Taylor Series and consider the following linear approximation around the point, $\hat{x}_{i+1}, \hat{f}_i = (\hat{x}_i, \hat{u}_i)$:

$$x_{i+1}^T f_i \approx -\hat{x}_{i+1}^T \hat{f}_i + \hat{x}_{i+1}^T f_i + x_{i+1}^T \hat{f}_i .$$

Then the Hamiltonian of the augmented problem takes the following form H^* for H^* :

$$\begin{aligned}
H^* &= \sum_{i=1}^N \phi_i(x_i, u_i) + K \sum_{i=1}^N [x_{i+1}^T x_{i+1} + f_i^T f_i + 2\hat{x}_{i+1}^T \hat{f}_i - 2\hat{x}_{i+1}^T f_i - 2x_{i+1}^T \hat{f}_i] \\
&\quad + K(x_1 - x_1^0)^T (x_1 - x_1^0) + \sum_{i=1}^N \lambda_{i+1}^T f_i + \lambda_1^T x_1^0 \\
&= \sum_{i=1}^N [\phi_i + Kf_i^T f_i + 2\hat{x}_{i+1}^T \hat{f}_i - 2\hat{x}_{i+1}^T f_i - 2x_{i+1}^T \hat{f}_{i-1} + \lambda_{i+1}^T f_i + Kx_i^T x_i] \\
&\quad + \lambda_1^T x_1^0 + Kx_{N+1}^T x_{N+1} - 2Kx_{N+1}^T \hat{f}_N + Kx_1^0 x_1^0 \\
&= \sum_{i=1}^N H_i^* + \lambda_1^T x_1^0 + Kx_{N+1}^T x_{N+1} - 2Kx_{N+1}^T \hat{f}_N + Kx_1^0 x_1^0 \tag{III-13}
\end{aligned}$$

where $\hat{f}_0 \equiv x_1^0$ and we have defined a stage Hamiltonian

$$H_i^* = \{\dot{x}_i + Kx_i^T x_i + Kf_i^T f_i + 2K\hat{x}_{i+1}^T \hat{f}_i - 2K\hat{x}_{i+1}^T f_i - 2x_{i+1}^T \hat{f}_{i-1}\} + \lambda_{i+1}^T f_i .$$

Let us now establish the following result which also constitutes the basis of the proposed algorithm (given later in this chapter).

Theorem 11: If the point $\tilde{u}^T = (\tilde{u}_1^T, \tilde{u}_2^T, \dots, \tilde{u}_N^T)$ minimizes the overall Hamiltonian H^* , then each stage Hamiltonian H_i^* , $i=1, \dots, N$, resulting after the Taylor Series linear approximation of the crossproduct terms is minimized with respect to the corresponding control variable u_i at the point \tilde{u}_i .

Proof: At the solution point we require feasibility, i.e.

$$\hat{x}_{i+1} = f_i(x_i, u_i) .$$

(This requirement is evident from the algorithm to be presented in the next section.) From eq. (III-13) it is clear then that

$$\frac{\partial H^*}{\partial u_i} = \frac{\partial H_i^*}{\partial u_i} = 0 \quad i=1, \dots, N,$$

and therefore a stationary point of H^* with respect to u_i is a stationary point of H_i^* with respect to u_i . The Hessian matrix of the second derivatives of H^* must be positive definite at the point \hat{u} , i.e.

$$A = \begin{bmatrix} \frac{\partial^2 H^*}{\partial u_1^2} & & & & & & \\ & 0 & & & & & \\ & & \frac{\partial^2 H^*}{\partial u_2^2} & & & & \\ & & & \frac{\partial^2 H^*}{\partial u_3^2} & & & \\ & & & & \ddots & & \\ & & & & & & \frac{\partial^2 H^*}{\partial u_N^2} \\ 0 & & & & & & \end{bmatrix}$$

must be positive definite. Thus, all the submatrices on the diagonal

must be positive definite, i.e., $\frac{\partial^2 H^*}{\partial u_1^2}$, $\frac{\partial^2 H^*}{\partial u_2^2}$, ..., $\frac{\partial^2 H^*}{\partial u_N^2}$ must be

positive definite. But $\frac{\partial^2 H^*}{\partial u_i^2} = \frac{\partial^2 H_i^*}{\partial u_i^2}$ therefore all the $\frac{\partial^2 H_i^*}{\partial u_i^2}$, $i=1,\dots,N$

are positive definite.

Theorem 11 implies that the minimization of the H^* with respect to the control variables can be replaced by the problems

$$\min_{u_i} H_i^* \quad i=1,\dots,N .$$

III.4. The Algorithm and Computational Characteristics

The theorems of section III.3. imply the following algorithm which constitutes a stronger version of the now available discrete minimum principle.

Step 1: Assume a value for the penalty constant K.

Step 2: Assume values for the control variables $\hat{u}_1, \dots, \hat{u}_N$.

Step 3: Using the values $\hat{u}_1, \dots, \hat{u}_N$, solve the state equations

$$x_1 = x_1^0$$

$$x_{i+1} = f_i(x_i, u_i) \quad i=1, \dots, N$$

forward and find x_2, \dots, x_N . Let $\hat{x}_2, \dots, \hat{x}_N$ be the found values ($\hat{x}_1 = x_1^0$, given constant).

Step 4: Using the values $\hat{u}_1, \dots, \hat{u}_N$ and $\hat{x}_1, \dots, \hat{x}_N$, solve the adjoint equations

$$\lambda_i = \frac{\partial \phi_i}{\partial x_i} + \frac{\partial f_i^T}{\partial x_i} \lambda_{i+1} \quad i=1,2,\dots,N$$

$$\lambda_{N+1} = 0$$

backwards. Let $\hat{\lambda}_1, \dots, \hat{\lambda}_{N+1}$ be the found values.

Step 5: Formulate the Hamiltonian H^* of the augmented problem (III-P2) and expand the crossproduct terms around the point

$(\hat{x}_1, \dots, \hat{x}_N; \hat{u}_1, \dots, \hat{u}_N)$. Formulate the stage Hamiltonian H_i^* , $i=1, \dots, N$. Minimize all H_i^* , $i=1, \dots, N$ and find optimal values for the controls, say \tilde{u}_i , $i=1, \dots, N$. If the minimization procedure fails to produce a minimum for at least one H_i^* increase K and go to step 2.

Step 6: If $|\tilde{u}_{ij} - \hat{u}_{ij}| < \epsilon$ for all j and for $i=1, \dots, N$ (j denotes the jth component of the vector u_i) stop, the solution is found, otherwise assume new values for the controls putting

$$\text{New } \hat{u}_i = \tilde{u}_i, \quad i=1, \dots, N$$

and go back to step 3.

The effective use of the above algorithm depends largely on the value of the constant K. Very small values of K will not produce the necessary convexity of the stage Hamiltonians, while a very large K will mask the objective function and will make the algorithm insensitive to the descent direction of the objective function. Miele et al. (1972)

have suggested a method for choosing a proper value for the constant K in the context of the Hestenes method of multipliers. As will be discussed in section III-5, the two methods are related and this method for choosing K should be satisfactory here.

III.5. Discussion and Conclusions

As mentioned before, the failure of the strong discrete minimum principle was caused by the fact that the stationary points of the stage Hamiltonian with respect to the controls are not always minima points. The method proposed in the previous sections of this chapter turns every stationary point of the Hamiltonian into a minimum point for a large enough K . Thus we can find the solution of the original problem at a local minimum point of the stage Hamiltonians. This constitutes a stronger result than that currently available, where we must search for a stationary point of the stage Hamiltonians.

The method used in this chapter to establish a stronger version of the discrete minimum principle parallels in many respects the method used in Chapter II to resolve the dual gaps in the two-level optimization procedure. The source of the shortcomings for both the methods (minimum principle and two-level method) is the nonconvexity of the objective function and/or the stage transformation equations. The two-level optimization method fails if either the objective function or transformation equations or both are nonconvex with respect to the control and/or the state variables.

For the resolution of the dual gaps of the two-level method, the same penalty term was used, multiplied by a positive constant K . It was required that K be large enough so that the stage sub-Lagrangians

become locally convex with respect to the control and state variables. Since, in the present work, we have required that the K be large enough to turn the stage Hamiltonians convex with respect to the controls only, we conclude that the K required by the strengthened form of the minimum principle to solve a nonconvex problem is at most as large as the K required by the two-level optimization method to resolve dual gaps.

Fig. 4 compares the values of K required to solve a nonconvex problem for the three methods discussed, namely, weak discrete minimum principle, the strengthened form of discrete minimum principle developed in this work, and the two-level optimization method. Note, $K_{TLM} \geq K_{SMP} \geq 0$, where K_{TLM} is the least value of K required for the two-level method and K_{SMP} is the least value of K for the minimum principle developed here. Thus it should be clear that, although the methods are related, they do not have equivalent shortcomings; it is possible to find problems which can be solved by the weak minimum principle or even the method developed here, and not by the two-level method. However given a $K \geq K_{TLM}$, all three methods succeed.

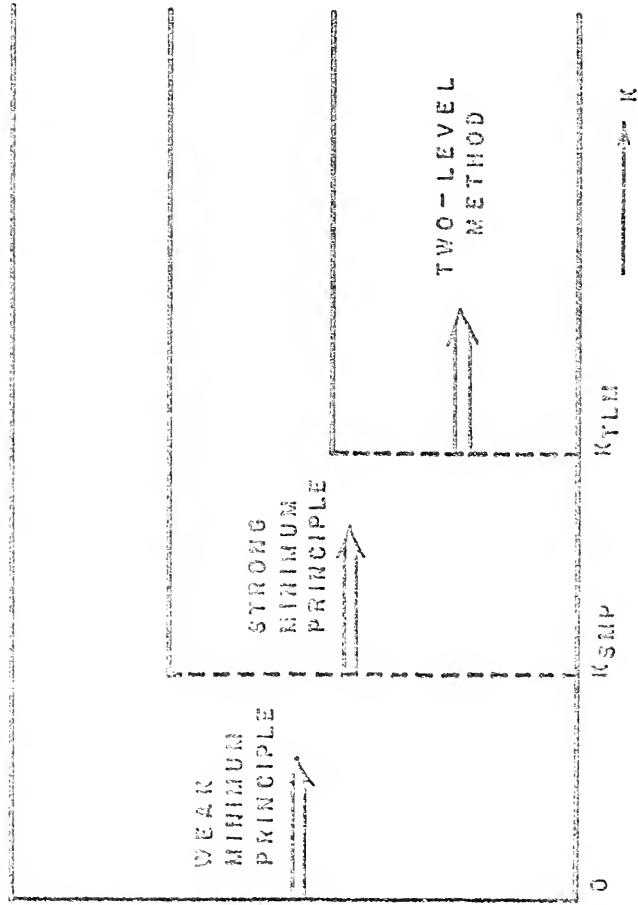


Figure 4. Relationship of Penalty Constant K Required for
Three Lagrangian Based Optimization Algorithms

CHAPTER IV
EXAMPLES ON NONCONVEX OPTIMIZATION

IV.1. Numerical Examples

IV.1.a. Two-stage Examples

Consider the following two stages example, Fig. 5, taken from McGalliard's thesis (McGalliard ,1971) and described by:

$$\text{Min } F = -t_3(x_2, u_2) + x_1^{0.6} + 2u_1 + x_2^{0.6} + 5u_2$$

$$\text{s.t. } x_2 = t_2(x_1, u_1) = 3x_1 + 3u_1$$

$$x_3 = t_3(x_2, u_2) = 2x_2 + 2u_2$$

$$S_1 = \begin{bmatrix} x_1 \geq 0 \\ u_1 \geq 0 \\ x_1 \leq 3 \\ x_1 + 2u_1 \leq 4 \end{bmatrix} \quad S_2 = \begin{bmatrix} x_2 \geq 0 \\ u_2 \geq 0 \\ u_2 \leq 1 \\ x_2 + 2u_2 \leq 4 \end{bmatrix}.$$

Using the regular two-level optimization procedure, a table is generated, see Table 1, and the solution is found to be:

$$x_1 = 0 \text{ or } 3 \quad x_2 = 4$$

$$u_1 = 0 \quad u_2 = 0$$

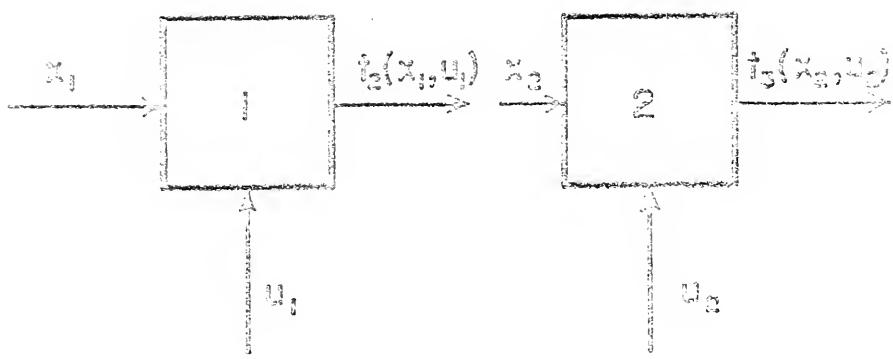


Figure 5. Two-stage Example

TABLE 1
POINTS GENERATED FOR THE TWO STAGE EXAMPLE

λ_2	x_1	u_1	x_2	u_2	Min ℓ_1	Min ℓ_2	$h(\lambda_2)$
$-\infty + \infty$	3	0.5	0	0	$-\infty$	0	$-\infty$
2	3	0.5	0	0	-18.07	0	-18.07
1.425	3	0.5	0.4	0	-12.1	0	-12.1
1.	3	0.5	4	0			- 9.3
0.667	3	0.5,0	4	0			- 7.1
0.5	3	0	4	0			- 6.3
0.215	3,0	0	4	0	0	-4.84	- 4.34
0.0	0	0	4	0	0	-5.7	- 5.7
-0.5	0	0	4	0	0	-7.7	- 7.7
$-\infty - \infty$	0	0	4	0	0	$-\infty$	$-\infty$

$$\text{and } \underset{\lambda}{\text{Max}} \underset{x_1, u_1, x_2, u_2}{\text{Min}} [F - \lambda(x_2 - 3x_1 - 3u_1)] = -4.84 .$$

Two solutions are produced with

$$z_1 = x_2 - 3x_1 - 3u_1 = +4 - 3(3) + 0 = -5$$

$$z_2 = x_2 - 3x_1 - 3u_1 = +4 - 3(0) + 0 = 4$$

and from the corollary in Section III.3. we conclude that a dual gap exists.

Introduce the penalty term $K(x_2 - 3x_1 - 3u_1)^2$ in the objective F

and make a linear approximation of the crossproduct terms x_2x_1 and

x_2u_1 . Following the algorithm in Section III.5., we find the solution:

$$x_1 = 4/3 \quad x_2 = 4 \quad \text{with } \lambda = 0.5 \text{ and } K = 10$$

$$u_1 = 0 \quad u_2 = 0 \quad \text{and}$$

$$\underset{x_1, u_1, x_2, u_2}{\text{Min}} F = \underset{\lambda}{\text{Max}} \underset{x_1, u_1, x_2, u_2}{\text{Min}} [F - \lambda(x_2 - 3x_1 - 3u_1)] = -4.51 .$$

Substitute unit 2, in example 1, by another described as follows:

$$f_2 = -t_3(x_2, u_2) + 2x_2^{0.6} + 2u_2$$

$$t_3(x_2, u_2) = 2x_2 + 3u_2$$

$$S_2 = \begin{bmatrix} x_2, u_2 \geq 0 \\ u_2 \leq 2 \\ x_2 + u_2 \leq 4 \end{bmatrix} .$$

The problem becomes:

$$\text{Min } F = x_1^{0.6} + 2u_1 - 2x_2 - u_2 + 2x_2^{0.6}$$

$$\text{s.t. } x_2 = 3x_1 + 3$$

$$x_1, u_1 \in S_1$$

$$x_2, u_2 \in S_2' .$$

The two-level procedure yields the solution (see Table 2)

$$\begin{aligned} x_1 &= 0 \quad \text{and} \quad 3 & x_2 &= 4 \quad \text{with} \quad \max_{\lambda} \min_{x_1, u_1, x_2, u_2} [F - \lambda(x_2 - 3x_1 - 3u_1)] = -2.54 \\ u_1 &= 0 & u_2 &= 0 \end{aligned}$$

and a dual gap is detected again.

The proposed method yields the solution

$$x_1 = 4/3 \quad x_2 = 4 \quad \text{for } \lambda = 0.8 \quad \text{and} \quad K = 10$$

$$u_1 = 0 \quad u_2 = 0$$

and

$$\min_{x_1, u_1, x_2, u_2} F = \max_{\lambda} \min_{x_1, u_1, x_2, u_2} [F - \lambda(x_2 - 3x_1 - 3u_1)] = -2.07$$

with all constraints satisfied.

IV.1.b. Three-stage example with recycle

In the two-stage example, let us insert an additional stage with a recycle loop from this new stage to stage 1, Fig 6.

TABLE 2
POINTS GENERATED FOR THE MODIFIED TWO-STAGE EXAMPLE

x_2	x_1	u_1	x_2	u_2	Min c_1	Min c_2	$h(x_2)$
$\rightarrow - \infty$	3	0.5	0	2	$\rightarrow - \infty$	-2.0	$\rightarrow - \infty$
3	3	0.5	0	2	-28.56	-2.0	-30.56
2	3	0.5	0	2	-18.06	-2.0	-20.06
1	3	0.5	0	2	-7.56	-2.0	-9.56
0.667	3	0.5,0	0	2	-4.06	-2.0	-6.06
0.5	3	0	0	2	-2.56	-2.0	-4.56
0.424	3	0	0,2	2	-2.42	-2.0	-4.42
0.3	3	0	2	2	-0.767	-2.37	-3.137
0.218	3	0	2,4	2,0	-0.031	-2.53	-2.561
0.215	3,0	0	4	0	0	-2.546	-2.546
0.1	0	0	4	0	0	-3.00	-3.00
$\rightarrow - \infty$	0	0	4	0	0	$\rightarrow - \infty$	$\rightarrow - \infty$

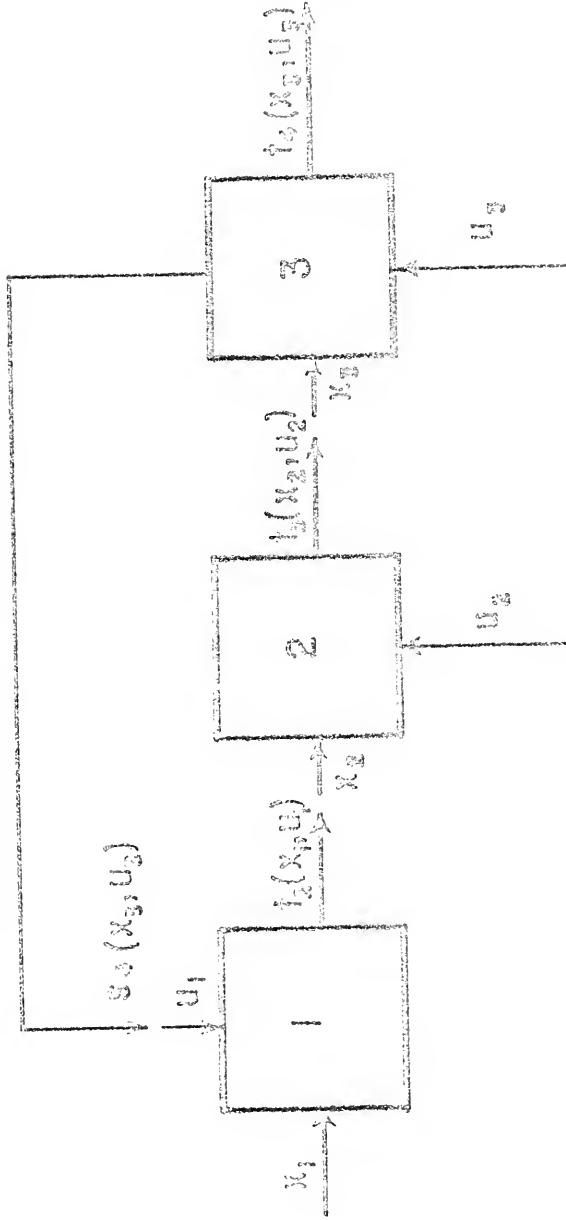


Figure 6. Three-stage Example with Recycle

The problem is:

$$\text{Min } F = x_1^{0.6} + 2u_1 + x_2^{0.6} + 5u_2 - 4x_3 - u_3 + x_3^{0.4}$$

s.t.

$$x_2 = 3x_1 + 3u_1$$

$$x_3 = 2x_2 + 2u_2$$

$$u_1 = u_3/4$$

$$x_1, u_1 \in S_1$$

$$x_2, u_2 \in S_2$$

$$x_3, u_3 \in S_3 = \begin{bmatrix} x_3, u_3 \geq 0 \\ x_3 \leq 4 \\ x_3 + u_3 \leq 6 \end{bmatrix}.$$

Since the solution of this problem is in a dual gap, we apply the proposed resolution which yields

$$x_1 = 0.67 \quad x_2 = 2 \quad x_3 = 4$$

$$u_1 = 0 \quad u_2 = 0 \quad u_3 = 0$$

for $\lambda_1 = 1.0 \quad \lambda_2 = 1.0, \quad \lambda_3 = 0.5 \quad \text{and} \quad K = 10.$

$$\begin{aligned} \text{Min } F &= \max_{\lambda_1, \lambda_2, \lambda_3} \min_{x_1, u_1, x_2, u_2, x_3, u_3} [F - \lambda_1(x_2 - 3x_1 - 3u_1) - \lambda_2(x_3 - 2x_2 - 2u_2) \\ &\quad - \lambda_3(u_1 - u_3/4)] \\ &= -11.96 \end{aligned}$$

with all constraints satisfied.

IV.1.c. Soland's example

This example is taken from Soland (1971):

$$\text{Min } f = -12x_1 - 7x_2 + x_2^2$$

$$\text{s.t. } \phi(x) = -2x_1^4 + 2 - x_2 = 0$$

$$0 \leq x_1 \leq 2$$

$$0 \leq x_2 \leq 3$$

The solution found by Soland (which is not the global minimum) after the generation of 12 points, which necessitated the solution of 23 subproblems is:

$$x_1 = 0 \quad \text{and} \quad f = -10$$

$$x_2 = 2$$

The algorithm proposed in this work generated a sequence of 8 points, solving 8 subproblems, and found the solution:

$$x_1 = 0.718 \quad \text{and} \quad f = -16.7387$$

$$x_2 = 1.46847$$

The sequence of generated points is shown in Table 3. The starting point is $\hat{x}_1 = 2.00$, $\hat{x}_2 = 0.0$.

IV.1.d. Jackson and Horn's counterexample

Consider the following two-stage example, Fig. 7, taken from Horn and Jackson (1965a). This example is the counterexample which demonstrated the fallacy of the strong minimum principle and is

TABLE 3
POINTS GENERATED FOR SOLAND'S EXAMPLE

	x_1	x_2	$h(x)$
-60	2.00	0.025	-919.04
-58.9	1.7365	0.0393	-713.07
-25.5	1.6513	0.06146	-182.64
-18.9	1.5511	0.07873	-108.35
-12.5	1.4318	0.1719	- 78.52
- 6.5	1.2799	0.3435	- 27.25
- 1.0	1.06	0.6326	- 22.16
÷ 3.0	0.718	1.46847	- 16.7389

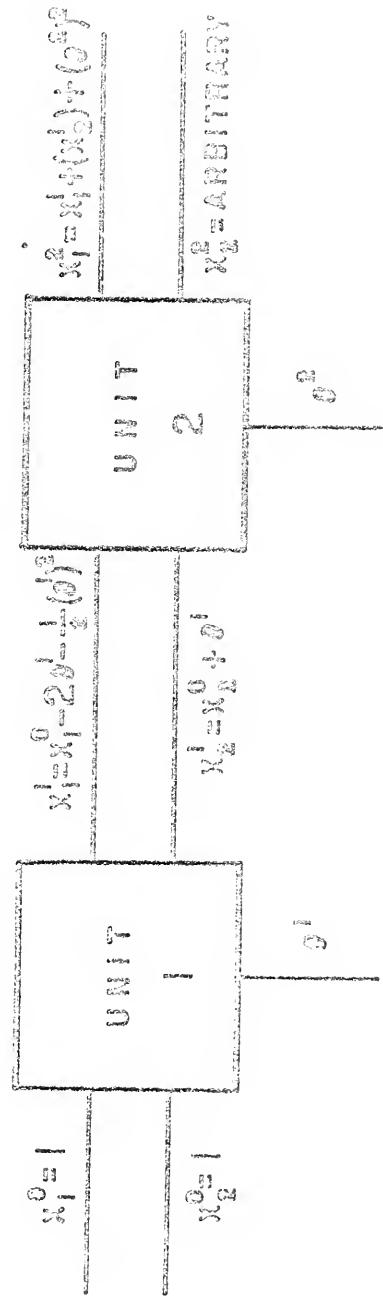


Figure 7. Jackson and Horn's Counterexample

described by:

$$\underset{\theta^1, \theta^2}{\text{Min}} \quad x_1^2$$

subject to

$$x_1^1 = x_1^0 - 2\theta^1 - \frac{1}{2}(\theta^1)^2$$

$$x_2^1 = x_2^0 + \theta^1$$

$$x_1^2 = x_1^1 + (x_2^1)^2 + (\theta^2)^2$$

$$x_2^2 = \text{arbitrary}$$

$$x_1^0 = x_2^0 = 1.$$

The solution to this problem is $\theta^1 = \theta^2 = 0$ which gives the smallest value of x_1^2 . It, however, does not minimize the Hamiltonian for the stage 1 since this Hamiltonian has only one stationary point which is a maximum. Let us now apply the procedure of the strong discrete minimum principle developed in the previous sections. Thus we have:

$$\begin{aligned} H_1^* &= K[x_1^0 - 2\theta^1 - \frac{1}{2}(\theta^1)^2]^2 + K[x_2^0 + \theta^1]^2 - 2K(\hat{x}_1^1)[x_1^0 - 2\theta^1 - \frac{1}{2}(\theta^1)^2] \\ &\quad - 2K(\hat{x}_2^1)[x_2^0 + \theta^1] + K(\hat{x}_1^1)[x_1^0 - 2\theta^1 - \frac{1}{2}(\theta^1)^2] + K(\hat{x}_2^1)[x_2^0 + \theta^1] \\ &\quad + \lambda_1^1 x_1^0 - 2\lambda_1^1 \theta^1 - \frac{1}{2}\lambda_1^1 (\theta^1)^2 + \lambda_2^1 x_2^0 + \lambda_2^1 \theta^1 \end{aligned}$$

and

$$\begin{aligned}
 H_2^* &= x_1^1 + (x_2^1)^2 + (\theta^2)^2 + K(x_1^1)^2 + K(x_2^1)^2 - 2K(x_1^1)[x_1^0 - 2\hat{\theta}^1 - \frac{1}{2}(\hat{\theta}^1)^2] \\
 &\quad - 2K(x_2^1)[x_2^0 + \hat{\theta}^1] + K(x_1^1)[x_1^0 - 2\hat{\theta}^1 - \frac{1}{2}(\hat{\theta}^1)^2] + K(x_2^1)[x_2^0 + \hat{\theta}^1] \\
 &\quad - \lambda_1^1 x_1^1 - \lambda_2^1 x_2^1
 \end{aligned}$$

with

$$\lambda_1^1 = 1 + 2K(x_1^1) - 2K[x_1^0 - 2\hat{\theta}^1 - \frac{1}{2}(\hat{\theta}^1)^2]$$

$$\lambda_2^1 = 2(x_2^1) + 2K(x_2^1) - 2K[x_2^0 + \hat{\theta}^1] .$$

The first necessary conditions yield:

$$\begin{aligned}
 \frac{\partial H_1^*}{\partial \theta^1} &= K(\theta^1)^3 + 6K(\theta^1)^2 + [10K - 2Kx_1^0 + 2K(\hat{x}_1^1) - \lambda_1^1]\theta^1 \\
 &\quad + [-4Kx_1^0 + 2Kx_2^0 + 4K(\hat{x}_2^1) - 2K(\hat{x}_2^1) - 2\lambda_1^1 + \lambda_2^1] = 0
 \end{aligned}$$

and

$$\frac{\partial H_2^*}{\partial \theta^2} = 2(\theta^2)^2 = 0 \quad \text{which gives } \theta^2 = 0 .$$

1. Put $K = 1$

$$2. \text{ Assume, } \hat{\theta}^1 = 1 \quad \hat{\theta}^2 = 0$$

$$3. \text{ Find, } x_1^1 = 1 - 2 - \frac{1}{2} = -\frac{3}{2} \quad \text{and} \quad x_2^1 = 1 + 1 = 2$$

$$4. \text{ Also, } \lambda_1^1 = 1 \quad \text{and} \quad \lambda_2^1 = 2(2) = 4$$

5. Minimize H_1^* and H_2^* which give: $\tilde{\theta}^1 = 0.947$ and $\tilde{\theta}^2 = 0$.

$$|\hat{\theta}^1 - \tilde{\theta}^1| = |.053| > \epsilon = 0.001$$

Go back to step 2,

2a. Assume $\hat{\theta}^1 = 0.947$ and $\hat{\theta}^2 = 0$

3a. Find, $x_1^1 = -1.343$ and $x_2^1 = 1.947$

4a. Find, $\lambda_1^1 = 1$ and $\lambda_2^1 = 3.894$

5a. Minimize H_1^* and H_2^* and find: $\tilde{\theta}^1 = 0.9$, $\tilde{\theta}^2 = 0$

Go back to step 2a and assume $\hat{\theta}^1 = 0.9$, $\hat{\theta}^2 = 0$. Continue in the same way until $|\hat{\theta}^1 - \tilde{\theta}^1| < \epsilon$ and $|\hat{\theta}^2 - \tilde{\theta}^2| < \epsilon$. This finally leads, in 7 iterations, to the solution $\theta^1 = \theta^2 = 0$ which is the solution of the problem.

Note that if $K = 0.1$ the nonconvexity of the first connection constraint with respect to θ^1 does not disappear. The stronger version of the minimum principle developed in this paper does not succeed in giving the solution.

IV.1.e. Denn's counterexample

Consider the following example taken from Denn (1969), which is also a counterexample to Katz' strong minimum principle and is described by:

$$\begin{array}{l} \text{Minimize } -x_1^2 \\ \text{u}_1^1, u_2^2 \end{array}$$

subject to

$$\dot{x}_1^1 = x_1^0(1 + x_2^0) - \frac{1}{2}(u^1)^2$$

$$\dot{x}_2^1 = 4x_1^0 - 2x_2^0 + u^1$$

$$\dot{x}_1^2 = x_1^1(1 + x_2^1) - \frac{1}{2}(u^2)^2$$

$$\dot{x}_2^2 = 4x_1^1 - 2x_2^1 + u^2$$

$$x_1^0 = -1/4$$

and

$$x_2^0 = 1$$

where u^1 and u^2 are to be chosen subject to:

$$0 < u^* \leq u^1, u^2.$$

This is a constrained problem but it can be handled equally as well as the unconstrained problem by the developed method. The inequalities will be handled through Kuhn-Tucker multipliers.

The above problem has a solution at the point $\tilde{u}_1 = 1$ and $\tilde{u}_2 = u^*$,

but the Hamiltonian for the first stage does not have a minimum and thus Katz' strong minimum principle cannot be used. In fact the Hamiltonian of the first stage possesses one stationary point which is a maximum. Let us now apply the algorithm developed in the present work. Thus we take:

$$\begin{aligned}
H_1^* = & - \frac{1}{2} \lambda_1^1 (u^1)^2 + (\lambda_2^1 + \mu_1) u^1 + \lambda_1^2 x_1^1 (1 + x_2^1) + 4\lambda_2^2 x_1^1 - 2\lambda_2^2 x_2^1 + K(x_1^1)^2 (1+x_2^1)^2 \\
& + 16K(x_1^1)^2 + 4K(x_2^1)^2 - 16Kx_1^1 x_2^1 - \mu_1 u^* + K\hat{x}_1^2 \hat{x}_2^1 (1 + \hat{x}_2^1) - 2K\hat{x}_1^2 x_1^1 (1+x_2^1) \\
& + \frac{K}{2} \hat{x}_1^1 (1 + \hat{x}_2^1) (\hat{u}^2) - Kx_1^1 (1 + x_2^1) (\hat{u}^2) + 4K\hat{x}_1^1 \hat{x}_2^2 - 8Kx_1^1 \hat{x}_2^2 \\
& - 2K\hat{x}_2^1 \hat{x}_2^2 + 4Kx_2^1 \hat{x}_2^2 - 4K\hat{x}_1^1 \hat{u}^2 + 8Kx_1^1 \hat{u}^2 + 2Kx_2^1 \hat{u}^2 - 4Kx_2^1 \hat{u}^2
\end{aligned}$$

and

$$\begin{aligned}
H_2^* = & - x_1^2 - \frac{1}{2} \lambda_1^2 (u^2)^2 + \lambda_2^2 u^2 + Kx_1^2 + \frac{K}{4} (u^2)^4 + Kx_1^2 (u^2)^2 + K(x_2^2)^2 + K(u^2)^2 \\
& - 2Kx_2^2 u^2 - \mu_1 (u^* - u^2) + K\hat{x}_1^2 \hat{x}_1^1 (1 + \hat{x}_2^1) - 2K\hat{x}_1^2 \hat{x}_1^1 (1 + \hat{x}_2^1) + \frac{K}{2} \hat{x}_1^1 (1 + \hat{x}_2^1) (\hat{u}^2) \\
& - K\hat{x}_1^1 (1 + \hat{x}_2^1) (u^2) + 4K\hat{x}_1^1 \hat{x}_2^2 - 8Kx_1^1 x_2^2 - 2K\hat{x}_2^1 \hat{x}_2^2 + 4K\hat{x}_2^1 x_2^2 \\
& - 4K\hat{x}_1^1 \hat{u}^2 + 8K\hat{x}_1^1 u^2 + 2K\hat{x}_2^1 \hat{u}^2 - 4K\hat{x}_2^1 u^2 .
\end{aligned}$$

The necessary conditions to be satisfied are:

$$\frac{\partial H_1^*}{\partial u^1} = - \lambda_1^1 (u^1) + \lambda_2^1 + \mu_1 = 0 \quad \text{and}$$

$$\begin{aligned}
\frac{\partial H_2^*}{\partial u^2} = & K(u^2)^3 - \lambda_1^2 (u^2)^2 + \lambda_2^2 + Kx_1^2 + K - 2Kx_2^2 + \mu_2 - K\hat{x}_1^1 (1 + \hat{x}_2^1) \\
& + 8K\hat{x}_1^1 - 4K\hat{x}_2^1 = 0 .
\end{aligned}$$

The adjoint variables satisfy the following equations:

$$\lambda_1^1 = \dot{x}_1^2(1+x_2^1) + 4\lambda_2^2 + 2K(x_1^1)(1+x_2^1)^2 + 32K(x_1^1) - 16Kx_2^1 - 2K\dot{x}_1^2(1+x_2^1)$$

$$- K(1+x_2^1)(\dot{u}^2) - 8K\dot{x}_2^2 - 8K\dot{u}^2 ,$$

$$\lambda_2^1 = \lambda_1^2 x_1^1 - 2\lambda_2^2 + 2K(x_1^1)^2(1+x_2^1) + 8K(x_2^1) - 16Kx_1^1 - 2K\dot{x}_1^2 x_1^1$$

$$- Kx_1^1 \dot{u}^2 + 4K\dot{x}_2^2 - 4K\dot{u}^2 ,$$

$$\lambda_1^2 = - 2x_1^2 + K + K(u^2) - 2K\dot{x}_1^1(1 + \dot{x}_2^1)$$

$$\lambda_2^2 = 2K(x_2^2) - 2Ku^2 + 4K\dot{x}_2^1 .$$

Note that in the above equations the variables μ_1 and μ_2 are Kuhn-Tucker multipliers through which we handle the inequalities

$u^1 \geq u^*$ and $u^2 \geq u^*$. For μ_1 and μ_2 we have $\mu_1, \mu_2 \leq 0$.

Starting with $K = 10$ and initial assumptions $u^1 = 2$ and $u^2 = 2$, we apply the same algorithmic procedure as presented in the previous example, and after 5 iterations we find the solution

$$\tilde{u}^1 = 1 \quad \text{and} \quad \tilde{u}^2 = u^* ,$$

which is the solution for the problem.

IV.2. The Design of a Heat Exchange Network

Avery and Foss (1971) have demonstrated that the method of two-level optimization may not be generally applicable to chemical process

design problems due to the mathematical character of commonly encountered objective functions. Since the cost functions used in chemical engineering design problems typically include a term

$x^{0.6}$ where x is the throughput in a unit, nonconvex problems arise.

The method for resolving the nonconvexities, as described in Chapter II, will be applied in this section to the design of a heat exchange network presented by Avery and Foss and possessing inherent nonconvexities.

The cold stream D of Figure 8 is to be heated from temperature T_o to T_p . Three hot streams having flowrates a, b, c and temperatures t_a, t_b, t_c may be used for heating; here they are considered to have sufficient heat capacity and availability to accomplish the task. The distribution of the total heat load among the three exchangers is to be accomplished at the minimum cost of equipment, which is related to the total heat transfer surface area A by

$$C = \sum_{i=1}^3 C_i = \sum_{i=1}^3 \gamma_i A_i^\alpha . \quad (\text{IV-1})$$

The parameters γ and α are positive constants; typically $\alpha = 0.6$.

It is easily demonstrated that this problem has a minimum cost solution and it is unique.

Let x_1 and x_2 be the enthalpies of stream D entering and leaving respectively, the first heat exchanger (see Figure 9). Similarly y_1 and y_2 are the enthalpies for the second exchanger and z_1 and z_2 for the third. Note that x_1 and z_2 are known since the flowrate and

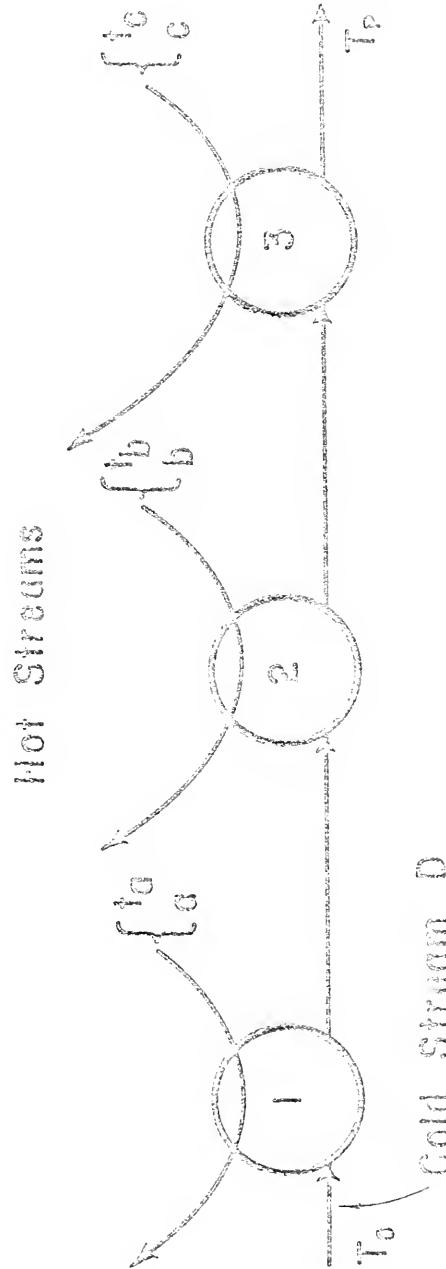


Figure 8. Heat Recovery Process

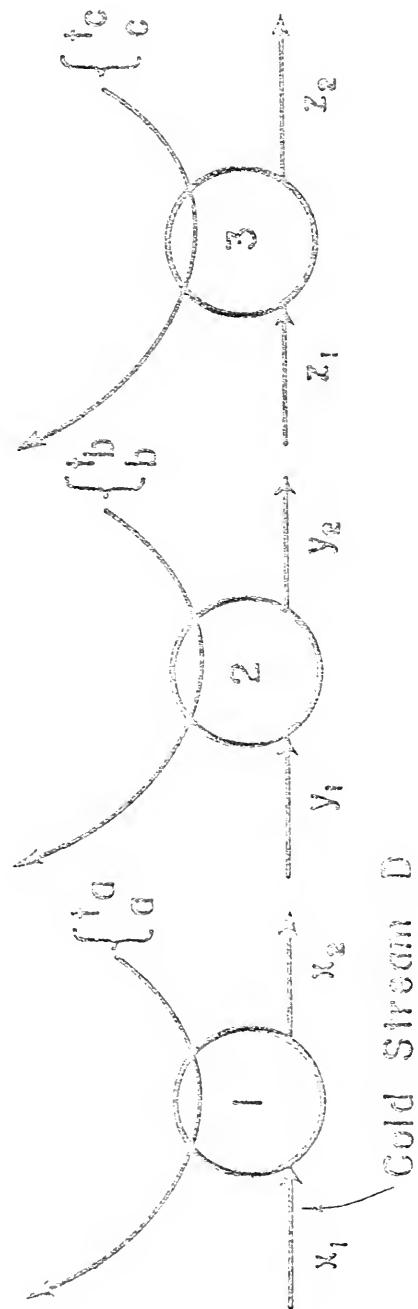


Figure 9. Uncoupled Heat Recovery Process Showing the Enthalpy Variables Used in the Two-level Optimization Method

temperatures of stream D are known at the entrance, to the first exchanger and the exit of the third exchanger. Thus, considering x_2, y_1, y_2 and z_1 as the design variables, the minimization problem can be formulated as follows:

$$\begin{aligned} \text{Min}_{x_2, y_1, y_2, z_1} \quad & [c_1(x_2) + c_2(y_1, y_2) + c_3(z_1)] \\ \text{s.t.} \quad & x_2 = y_1 \\ & y_2 = z_1 . \end{aligned} \quad (\text{IV-2})$$

The Lagrangian for this problem is:

$$\begin{aligned} L = & \sum_{i=1}^3 c_i - \lambda_1(x_2 - y_1) - \lambda_2(y_2 - z_1) \\ = & \{c_1(x_2) - \lambda_1 x_2\} + \{c_2(y_1, y_2) + \lambda_1 y_1 - \lambda_2 y_2\} + \{c_3(z_1) + \lambda_2 z_1\} \\ = & \ell_1(x_2, \lambda_1) + \ell_2(y_1, y_2, \lambda_1, \lambda_2) + \ell_3(z_1, \lambda_2) . \end{aligned} \quad (\text{IV-3})$$

For given values of the multipliers λ_1 and λ_2 , the two-level optimization method requires the minimization of the sublagrangians ℓ_1 , ℓ_2 and ℓ_3 .

A sufficient condition for a function to be at a minimum point is that the Hessian matrix be positive definite at that point. The Hessian for ℓ_2 is defined as

$$H_2 = \begin{bmatrix} \frac{\partial^2 c_2}{\partial Q^2} & \frac{\partial^2 c_2}{\partial Q \partial y_1} \\ \frac{\partial^2 c_2}{\partial y_1 \partial Q} & \frac{\partial^2 c_2}{\partial y_1^2} \end{bmatrix} = \begin{bmatrix} d_1 & e \\ e & d_2 \end{bmatrix} \quad (\text{IV-4})$$

and the conditions assuring that both eigenvalues are positive are

$$d_1 > 0 \quad , \quad d_2 > 0 \quad (\text{IV-5})$$

$$\Delta \equiv (d_1 d_2 - e^2) > 0 . \quad (\text{IV-6})$$

In the above, the heat duty $Q = y_2 - y_1$ in exchanger 2 has been substituted for the variable y_2 .

d_1 , d_2 and e may be determined from equation (IV-1), and the following relations, which are suitable for the determination of heat exchange area. Figure 10 identifies the notation used here.

$$Q = C_D (T_2 - T_1) = C_B (t_2 - t_1)$$

$$Q = y_2 - y_1$$

$$T_1 = y_1 / C_D$$

$$A = \frac{Q}{U \Delta T_m}$$

$$\Delta T_m = \frac{(t_2 - T_2) - (t_1 - T_1)}{\ln \frac{(t_2 - T_2)}{(t_1 - T_1)}} , \quad r \equiv \frac{C_D}{C_B} \neq 1$$

Without loss of generality, it is convenient to take $r > 1$; identical conclusions hold when $r \leq 1$. It may be shown that

$$d_1 = q[(\alpha-1)(rs-1)^2 + p\{(rs)^2 - 1\}]$$

$$d_2 = q[(\alpha-1)(s-1)^2 + p(s^2-1)]$$

$$e = q[(\alpha-1)(rs-1)(s-1) + p(rs^2-1)]$$

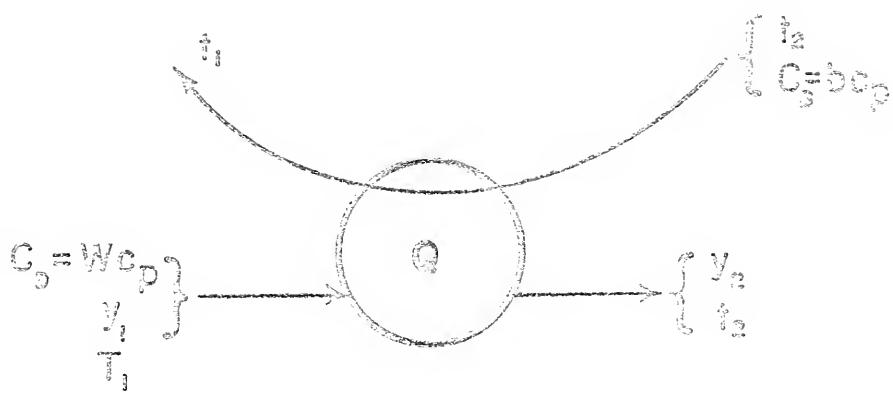


Figure 10. Diagram Showing Notation for a Single Exchanger

where

$$s = \frac{\delta_2}{\delta_1} = \frac{c_D t_2 - Q - y_1}{c_D t_2 - rQ - y_1} > 1$$

$$p = \ln \frac{\delta_2}{\delta_1} > 0$$

$$q = K \alpha p^{\alpha-2}/\delta_2^2 \quad 0$$

$$K = Y \left[\frac{c_D}{U(r-1)} \right]^\alpha > 0 .$$

Substitution of d_1 , d_2 and e into equation (IV-6) gives

$$\Delta \equiv (d_1 d_2 - e^2) = -q^2 p^2 s^2 (r-1)^2$$

which implies that Δ is never positive, and that, regardless of the parameters of the problem, the stationary point can never be a minimum. Therefore the two-level method fails.

Consider now the following augmented problem:

$$\text{Min } F^* = c_1(x_2) + c_2(y_1, y_2) + c_3(z_1) + K^*(x_2 - y_1)^2 + K^*(y_2 - z_1)^2$$

$$\text{s.t. } x_2 - y_1 = 0 \quad \text{and} \quad y_2 - z_1 = 0$$

with $K^* > 0$.

The coupled terms $x_2 y_1$ and $y_2 z_1$ are expanded in a Taylor Series around the point $\tilde{x}_2, \tilde{y}_1, \tilde{y}_2, \tilde{z}_1$ and are approximated by the linear terms in the expansion. The Lagrangian of the above problem becomes:

$$\begin{aligned}
L^* &\approx \{C_1(x_2) + (\lambda_1 + 2K^* \tilde{y}_1)x_2 + K^* x_2^2 + K^* \tilde{x}_2 \tilde{y}_1\} \\
&+ \{C_2(y_1, y_2) + (\lambda_1 - 2K^* \tilde{x}_2)y_1 + (\lambda_2 + 2K^* \tilde{z}_1)y_2 \\
&+ K^* y_1^2 + K^* y_2^2 + K^* \tilde{x}_2 \tilde{y}_1 + K^* \tilde{y}_2 \tilde{z}_1\} \\
&+ \{C_3(z_1) + (\lambda_2 - 2K^* \tilde{y}_2)z_1 + K^* z_1^2 + K^* \tilde{y}_2 \tilde{z}_1\} \\
&= \ell_1^* + \ell_2^* + \ell_3^* .
\end{aligned}$$

The necessary conditions for the minimum of C are

$$\nabla_v L^* = 0 \quad \text{and} \quad \nabla_\lambda L^* = 0$$

where

$$v = [x_2, y_1, y_2, z_1]^T \quad \text{and} \quad \lambda = (\lambda_1, \lambda_2)^T.$$

This results in the following equations

$$\frac{\partial \ell_1^*}{\partial x_2} = 0 , \quad \frac{\partial \ell_2^*}{\partial y_1} = \frac{\partial \ell_2^*}{\partial y_2} = 0 , \quad \frac{\partial \ell_3^*}{\partial z_1} = 0$$

$$x_2 = y_1 \quad \text{and} \quad y_2 = z_1 ;$$

that is, each subsystem is at a stationary point with respect to the variables associated with it. The Hessian matrix for the new sub-lagrangian ℓ_2^* of the second exchanger is:

$$H_2^* = \begin{bmatrix} \frac{\partial^2 C_2}{\partial Q^2} + 2K^* & \frac{\partial^2 C_2}{\partial Q \partial y_1} + 2K^* \\ \frac{\partial^2 C_2}{\partial y_1 \partial Q} + 2K^* & \frac{\partial^2 C_2}{\partial y_1^2} + 4K^* \end{bmatrix} = \begin{bmatrix} d_1^* & e^* \\ e^* & d_2^* \end{bmatrix} .$$

The terms d_1^* , d_2^* and e^* are easily found to be

$$d_1^* = d_1 + 2K^* , \quad d_2^* = d_2 + 4K^*$$

$$\text{and } e^* = e + 2K^* .$$

Then

$$\begin{aligned} \Delta^* &= [d_1^* d_2^* - (e^*)^2] \\ &= 4(K^*)^2 + (4d_1 + 2d_2 - 4e) K^* + (d_1 d_2 - e^2) \\ &= 4(K^*)^2 + \Gamma_2 K^* + \Gamma_3 \end{aligned}$$

where

$$\Gamma_2 = 4d_1 + 2d_2 - 4e$$

and

$$\Gamma_3 = d_1 d_2 - e^2 .$$

Δ^* has roots

$$K^* = \frac{-\Gamma_2 \pm (\Gamma_2^2 - 16\Gamma_3)^{\frac{1}{2}}}{8} .$$

It was shown before that $\Gamma_3 \leq 0$

thus

$$\left| (\Gamma_2^2 - 16\Gamma_3)^{\frac{1}{2}} \right| \geq |\Gamma_2|$$

and the roots are real and of opposite sign. Let $K_1^* \geq 0$ and $K_2^* \leq 0$, then for $K^* > K_1^*$, $\Delta^* > 0$. Also for

$$K^* > -d_1/2 \equiv K_3^* , \quad d_1^* > 0$$

and for

$$K^* > -d_2/2 \equiv K_4^* , \quad d_2^* > 0 .$$

Therefore if $K^* = \max \{K_1^*, K_3^*, K_4^*\}$ we find $H_2 > 0$ and the stationary point for unit 2 is a minimum as required.

Thus we see that the curvature of a sublagrangian at the stationary point can be determined not only by the sub-unit cost function, C_2 in this case, but also by the numerical value of the penalty factor K^* .

The above example demonstrates the applicability of the developed strategy in a wide class of chemical engineering design problems where the conventional two-level method fails to provide the solution.

CHAPTER V

THE SYNTHESIS OF PROCESS FLOWSHEETS. A GENERAL REVIEW.

Processing systems are characterized by two distinct features. The first is the nature of the process units and their interconnections, and the second is the capacities and the operating conditions of the units. Synthesis of optimal processing schemes requires a search over the whole space of structural alternatives as well as over the space of the design variables for each structural configuration. The synthesis of an optimal process flowsheet represents a great challenge for a chemical engineer in the area of systems engineering.

A number of significant contributions have moved the problem from the state of art to the state of semi-art where the personal capabilities of the designer have been organized and systematized in a logical manner. The size of the problem is very large and the methods currently available simply give good solutions which are happily accepted by practicing designers. The progress of mathematical systems theory in this area has been very slow and no major thrusts have been attained. Thus the chemical engineer has to resort to techniques that are available for use with his own experience and intuition.

We shall attempt to present the major contributions to the solution of the synthesis problem in order to relate and compare the synthesis strategies proposed in this work, and presented in Chapters VI and VII, with the general trends of thought already established by other researchers in this area.

Rudd (1968) proposed an approach to system synthesis based on process design decomposition whereby a design problem for which no previous technology existed is broken down into a sequence of sub-design problems until the level of the available technology is reached. Thus the original problem is decomposed into two smaller problems S_I and S_{II} which are united through a set of artificially imposed tear constraints. This decomposition continues until existing technology for the subtasks is reached. The selection among the alternate structures arising from the decomposition of a task into subtasks was made by comparing the economic performance of the structures generated. The economic performance of subtasks of unknown technology was estimated by an iterative scheme starting with a first approximation of the system's performance based on previously solved, similar design problems. In a second step Masso and Rudd (1969) used heuristics (rules of thumb) to define the points of decomposition and the set of artificially imposed tear constraints. The heuristics, rules based on the previous experience, ingenuity and intuition of the designer and updated through experience with the current problem, speed up the synthesis process, but they are fallible and they do not guarantee an optimal solution. This procedure was applied without complete success to the synthesis of heat exchange networks. The usefulness of a branch and bound strategy in screening the various structural alternatives, in conjunction with the above strategy, was examined by Lee, Masso and Rudd (1970). The design decomposition supported by heuristic decisions came to its advanced state with the works of Siirola and Rudd (1971) and Siirola, Powers and Rudd (1971). It was applied to the invention of chemical process

flowsheets. The synthesis of the flowsheet starts at its very heart, the chemical reactor. Reactor support systems such as preparation of the feed, separation of the products, purification, heating or cooling the feed or the products are identified. The reasonableness of the various routings for the components through the subunits identified is tested according to a set of process heuristics while the overall material flow is selected using linear programming to maximize the sum of heuristic scores from all the feasible routings. This strategy enables the synthesis of complicated integrated systems and has been applied to a process for the production of monochlorodecone by direct chlorination. Such an approach which relies heavily on heuristics can only be successful if satisfactory heuristics can be developed. Nevertheless it produces good solutions quickly.

In the electrical engineering literature a general approach is to postulate a network which must contain the desired optimum network as a subset of the one postulated. A similar approach has been followed by various researchers for the synthesis of chemical process flowsheets. The structure of a processing system is determined by a connection matrix with elements expressing the interconnections between each pair of subsystems. From this point of view the optimal synthesis of a chemical process flowsheet is to determine the optimal connection matrix expressing the interconnections among the subsystems with the satisfaction of the optimal design conditions. Thus integrated complex systems are first generated where all the available subsystems are included. Split fractions which are defined on the fraction of the output from one unit used as an input to another unit are assigned to each stream connecting two units and the optimal flowsheet is

generated by a direct optimization of the initial complex system, with respect to the design variables of each unit and the split fractions for each stream. The various methods which follow the above strategy differ in the way they tackle this large-scale optimization problem. Umeda and Ichikawa (1972) use an infeasible method through the two-level optimization procedure. Osakada and Fan (1973) employ a multilevel technique which at the first level minimizes the objective functions of the separated noninteracting subsystems, while at the second level the solutions of the subproblems are coordinated to render the optimum of the overall system. Umeda, Hirai and Ichikawa (1972) optimize the initial, large interconnected system using the complex algorithm. Umeda, Shindo and Tazaki (1972) have employed a feasible decomposition method at two levels. Ichikawa and Fan (1973) use a strategy based on the weak discrete minimum principle which results in a feasible decomposition. As an application of the theory of Chapter II of the present work, a variation of the two-level method was proposed (Stephanopoulos and Westerberg, 1974) to account for the nonconvexities of chemical engineering cost functions or connection constraints. Any method following the above general approach has to solve a large optimization problem, and, only when combined with heuristics (to disregard certain connections) and efficient decomposition techniques, can they be valuable.

A third general approach is that of the evolutionary procedure. This has been probably the method most likely to be used by a process designer. The evolutionary procedure consists of a finite sequence of structural changes each of which leads to a better flowsheet.

This procedure ends when no flowsheet can be found which can be generated from the last flowsheet by simple, one-step structural changes, and will also appreciate a lower cost. Then we say that we have found a "local minimum". Thus a processing system is devised, analyzed and changed in one or more ways so as to improve it. These steps are repeated until no further improvement in the flowsheet can be made. Such a strategy maintains the good points of the current flowsheet while the less promising parts of the flowsheet are changed in an effort to improve the overall process. This procedure resembles the hill climbing search techniques used by optimization algorithms to maximize a function with continuous variables. In the synthesis problem we have to deal with discrete variables which can change drastically from one flowsheet to another. As is the case with the optimization of functions with continuous variables, the optimum reached by an evolutionary synthesis is a "local optimum" unless the cost function over the various processes is in some sense unimodal. Furthermore repeated application of an evolutionary scheme from very different starting flowsheets will reveal more than one local optima. Also as King et al. argued, "It is questionable whether one can speak of a global optimum design for an open-ended process synthesis problem,..."

King, Gantz and Barnes (1972) applied this strategy as a succession of alterations on that portion of the most current flowsheet which could be changed for the greatest advantage. In the same work two examples of evolutionary synthesis were given. In the first example, the synthesis of a demethanizer tower in an ethylene plant, the evolution steps were identified by the designer based on his previous

experience in this particular processing area. In the second example the computer was used to implement the logic of the evolution steps. In both examples the identification of the portion of the process to be altered was guided by heuristics, while a second level of heuristics was used to guide the selection of the parts to replace the parts to be changed. This approach, being highly heuristic, cannot guarantee the optimum solution, but it can "help to structure the thinking of the design engineer" (King, Gantz and Barnes, 1972) and thus produce good solutions.

McGalliard and Westerberg (1972) have proposed an algorithmic approach to screen the alternatives developed quickly without necessitating the optimization of the entire process. It is based on Lasdon's two-level optimization method and the bounding characteristics of the dual and primal functions. They also pointed out that the basic structural modifications to a certain process can be to replace a unit with a different one, to delete a unit and to generate a recycle. Thus, since the change is local, the sensitivity information from the current structure is presented via the stream prices (Lagrange multipliers).

Pho and Lapidus (1973) have presented a strategy which generates systematically all possible heat exchange networks for problems of the type defined earlier by Rudd and coworkers. Their work can also be viewed as an evolutionary strategy, where the system continuously evolves to better systems in an organized manner. Thus for the synthesis of a heat exchange network, starting from a situation with no matches between the streams in question, all alternate first matches are examined. A set of structural changes is then available.

The most promising is retained and the rest are rejected. In this fashion the synthesis continues in an evolutionary sense with a simultaneous screening of the various alternatives open at a certain point. The evaluation of the alternatives is heuristic and follows the lookahead strategy used in chess.

For the solution of particular synthesis problems, specialized techniques have been developed, in addition to the general strategy mentioned above. Thus dynamic programming in conjunction with heuristics was used to synthesize separation schemes (Hendry and Hughes, 1972), reaction paths (Powers and Jones, 1973), reactor networks (Aris, 1964) and heat exchange networks (Westbrook, 1961). Hendry, Rudd and Seader (1973) presented a detailed account of all the specialized methods developed to synthesize such particular problems as the synthesis of heat exchange networks, separation trains, reaction paths, reactor networks, etc.

CHAPTER VI

BRANCH AND BOUND STRATEGY WITH LIST TECHNIQUES FOR THE SYNTHESIS OF OPTIMAL SEPARATION SCHEMES

Multicomponent separations arise very frequently in chemical processing systems, and so techniques for the synthesis of multi-component separation sequences play an important part in the synthesis of optimal processing schemes.

In this chapter we intend to develop a scheme that will enable us to relax the most of the restrictions that the methods already proposed include inherently, while retaining their attractive characteristics. Thus section VI.1. presents the previous works on the synthesis of optimal separation schemes, while the section VI.2 states the problem and develops the list techniques for the representation of separation trains. Section VI.3 develops the branch and bound strategy while section VI.4 presents two illustrative examples of the proposed strategy. Section VI.5 summarizes and discusses the proposed method.

VI.1. Previous Works on the Synthesis of Separation Schemes

With respect to the synthesis of multicomponent separation schemes, two developments are the most significant, the works by Thompson and King (1972) and by Hendry and Hughes (1972). These works represent very distinctly the general trends and attitudes in the synthesis area and, because of their relevance to the present work, will be examined further in some detail.

The approach proposed by Thompson and King (1972) is highly heuristic in nature. The first step is to identify a feasible set of products for the process. The synthesis begins with the choice of an initial separator. All possible first separations are identified and the cost for each one is estimated. These estimation parameters are updated during the synthesis. The cheapest is selected as the first separator and is rigorously costed. This procedure is repeated to choose a second separator then a third and so forth until no further separators are needed. With an updating of the cost estimation parameters the entire procedure is repeated until no change is found in the process. The identification of alternatives and their selection is guided by heuristics. After the separation sequence has been established, a new one is produced based on a new set of products and this continues until no lower cost process can be found.

For the prediction of the cost of a unit separating product i from product j using method k, estimation parameters β_{ijk} are introduced. The estimated cost of a separator is quite simply $(\beta_{ijk}) (N_{ijk})$, where N_{ijk} = number of stages required for the above separation. Initially all the β 's take the low value of \$1.0/stage in order to permit the consideration of all the viable alternatives. When a separation unit is selected, its β_{ijk} is updated using the real cost of the separator:

$$\beta_{ijk} = \text{real cost of separator} / (N_{ijk})$$

This approach being almost entirely heuristic cannot guarantee the optimality of the separation sequences derived. Its success in producing good solutions quickly depends largely on the success of the heuristics used.

Hendry and Hughes (1972) developed a strategy based on the use of dynamic programming. Unlike the previous method, this one is basically an algorithmic approach. In the first step all the two-components separations are evaluated, e.g.

$$\frac{A}{B}, \quad \frac{B}{C}, \quad \frac{C}{D}, \quad \frac{D}{E} \quad \text{etc.}$$

The generation of all the three-components separations follows, e.g.

$$\begin{array}{ccccc} \underline{A} & A & \underline{B} & B & \text{etc.} \\ B & \underline{\frac{B}{C}} & C & \underline{\frac{C}{D}} & \\ C & & D & & \end{array}$$

Since the cost for the separation of the remaining two components is known from the first step, the optimal sequence for the separation of three components into three products is easily derived. This procedure can continue up to the number of components in the original mixture. In fact this approach is an exhaustive enumeration over the set of all the possible separators, which incidentally is much smaller than the set of all possible flowsheets. As will be shown later the above procedure can be improved and a smaller class of separators will need to be examined. This approach, employing in fact the principle of dynamic programming, cannot be applied in cyclic systems, and it requires large computation time. Furthermore we must assume that essentially perfect splits take place to permit an analysis of a unit without consideration of the units preceding it and that mass separating agents will have to be recovered in the immediate successor unit. In this chapter we intend to develop a scheme that will enable us to relax most of the above restrictions and assumptions while retaining the attractive characteristics of the various methods.

VI.2. Statement of the Problem and the List Techniques for the Representation of the Separation Operations

The synthesis of separation sequences is a basic subproblem in the overall synthesis of chemical processing systems.

The general multicomponent separation problem can be stated as follows: "Select the proper separations (separation method and point of split) and their sequencing in order to minimize the overall venture cost of a separation process, which separates a mixture of N components into a prespecified number of product streams, using S different separation methods."

The maximum number of the alternative processing schemes which can be derived for the above problem is very large and is given by the following formula (Thompson and King, 1972)

$$\frac{(2(N-1))!}{N!(N-1)!} S^{N-1}$$

Consider the following mixture of five components A,B,C,D,E which must be separated into its five pure components. The first question we are confronted with is: what separation methods will be considered and how many. The answer to this comes from a close examination of the physical properties of the components, trying to exploit differences in their physical properties, and from the engineering judgement of the designer in conjunction with the fact that, with an increase in the number of the separation methods considered an immense increase in the number of the plausible alternatives is caused. Suppose that two methods are selected, method α and method β . The ranked lists for the two methods are formed. The ranked list is a list of the components of the initial mixture, in

decreasing order with respect to a physical property exploited by a certain separation method (e.g., boiling point for distillation, solubility for extraction, size of solid particles for screening).

Let the ranked lists for methods α and β be:

RL(1) : A,B,C,D,E

and

RL(2) : B,D,A,C,E

In developing the separation train two main problems arise: which separation method will be selected and where will the split be performed. Thus, using method α the following separators can be generated at the beginning of the separation train:

A	A	A	A
<u>B</u>	<u>B</u>	<u>B</u>	<u>B</u>
C	<u>C</u>	<u>C</u>	<u>C</u>
D	D	<u>D</u>	<u>D</u>
E	E	E	<u>E</u>

It is obvious that by selecting one of the above separators as the first one the consequent separators have been restricted to a more specified class of separators. Thus, a decision made at the first separator is going to affect the performance of the entire process not only by the performance of the first stage itself but also by the restrictions that the first decision imposes on the consequent separators. The answer to this dilemma will be given later by the approach developed.

Hendry and Hughes (1972) have proposed the use of list techniques to represent a separation operation. The basis for such a development is the ranked lists. Forming the ranked lists of the components

for the separation methods considered by the designer, all possible combinations of components which may be formed during the separation process may be derived. Furthermore all the distinct separators which may be part of a plausible flowsheet can be derived. This result follows from a consideration of the separation operations as "ranked list splitters" (Hendry and Hughes, 1972) which takes the ranked list and splits it into two adjacent sections. The components in each of these sections must then undergo further processing, perhaps using a different separation method, until the prespecified products are produced. Thus, the ranked lists define the various ways in which the separations may take place.

The following fact should be pointed out: the number of distinct separators which are generated from the ranked lists for a mixture of N components using S separation methods, is much smaller than the number of flowsheets the separators can generate when they are interconnected to each other to form a feasible process. In the Table 4, the corresponding numbers for different mixtures are shown with only one separation method considered. As N increases the following limits are approached:

$$\frac{F(N+1)}{F(N)} \longrightarrow 4 \quad \text{and} \quad \frac{B(N+1)}{B(N)} \longrightarrow 1$$

For $N \geq 3$ and $S > 2$ the number of distinct flowsheets is much larger than the number of distinct separators. Thus working within the set of distinct separators is an advantage since the search space is smaller.

Based on the ranked lists for all the separation methods considered, we can develop all the distinct separators which will

TABLE 4

THE NUMBERS OF DISTINCT SEPARATORS $B(N)$
AND DISTINCT FLOWSHEETS $F(N)$ FOR A MIXTURE
OF N COMPONENTS AND ONE SEPARATION METHOD

<u>Number of Components in a Mixture (N)</u>	<u>Number of Distinct Separators ($B(N)$)</u>	<u>Number of Distinct Flowsheets ($F(N)$)</u>
2	1	1
3	4	2
4	10	5
5	20	14
6	35	42
7	56	132
:	:	:
10	165	4,862
11	220	16,796

constitute the elements of every feasible process. First all the N -component separators are generated for every ranked list. This step produces $N-1$ different separators for every separation method considered and groups of components containing from one up to $N-1$ components. At the second stage all the $(N-1)$ or fewer component separators are generated based on the component groups derived at the first step which in turn produce smaller size groups. This procedure continues until all the n -component separators ($n=2,3,\dots,N$) have been generated. In the Figure 11, all the distinct separators for a 4 component mixture are shown with two separation methods α and β considered.

Hendry and Hughes (1972) have discussed in detail the various characteristics and consequences of the use of ranked lists. Thus for components with very close boiling points or solubilities in a certain solvent,etc., it may be reasonable that they be ranked equally thus implying that they cannot be separated using that method. Similarly, it often happens that a particular separation technique cannot be applied when certain components are present in the mixture. Consequently, the ranked list should not contain those components.

After using the ranked lists to generate all possible separators which will be part of any feasible separation scheme, we shall proceed to develop, in the next section, a strategy which allows the generation of a very small number of "nearly optimum" flowsheets or when carried even further, to generate the globally best flowsheet subject to the assumptions defining the problem.

4 - Component Separators

1. $\left(\begin{matrix} A \\ B \\ C \\ D \end{matrix}\right)_\alpha$

2. $\left(\begin{matrix} A \\ B \\ C \\ D \end{matrix}\right)_\beta$

3. $\left(\begin{matrix} A \\ B \\ C \\ D \end{matrix}\right)_\alpha$

4. $\left(\begin{matrix} B \\ A \\ C \\ D \end{matrix}\right)_\beta$

5. $\left(\begin{matrix} B \\ A \\ C \\ D \end{matrix}\right)_\beta$

6. $\left(\begin{matrix} B \\ A \\ C \\ D \end{matrix}\right)_\beta$

3 - Component Separators

7. $\left(\begin{matrix} B \\ C \\ D \end{matrix}\right)_\alpha$

8. $\left(\begin{matrix} B \\ C \\ D \end{matrix}\right)_\alpha$

9. $\left(\begin{matrix} B \\ C \\ D \end{matrix}\right)_\beta$

10. $\left(\begin{matrix} B \\ C \\ D \end{matrix}\right)_\beta$

11. $\left(\begin{matrix} A \\ B \\ C \end{matrix}\right)_\alpha$

12. $\left(\begin{matrix} A \\ B \\ C \end{matrix}\right)_\alpha$

13. $\left(\begin{matrix} B \\ A \\ C \end{matrix}\right)_\beta$

14. $\left(\begin{matrix} B \\ A \\ C \end{matrix}\right)_\beta$

15. $\left(\begin{matrix} A \\ B \\ C \end{matrix}\right)_\alpha$

16. $\left(\begin{matrix} A \\ B \\ C \end{matrix}\right)_\alpha$

17. $\left(\begin{matrix} A \\ B \\ C \end{matrix}\right)_\beta$

18. $\left(\begin{matrix} A \\ B \\ C \end{matrix}\right)_\beta$

2 - Component Separators

19. $\left(\begin{matrix} A \\ B \end{matrix}\right)_\alpha$

20. $\left(\begin{matrix} B \\ A \end{matrix}\right)_\beta$

21. $\left(\begin{matrix} B \\ C \end{matrix}\right)_\alpha$

22. $\left(\begin{matrix} B \\ C \end{matrix}\right)_\beta$

23. $\left(\begin{matrix} C \\ D \end{matrix}\right)_\alpha$

24. $\left(\begin{matrix} C \\ D \end{matrix}\right)_\beta$

25. $\left(\begin{matrix} A \\ C \end{matrix}\right)_\alpha$

26. $\left(\begin{matrix} A \\ C \end{matrix}\right)_\beta$

Figure 11. All the Distinct Separators Generated and the Basic Flowsheet for a Fictitious Example of 4 Components Using 2 Separation Methods

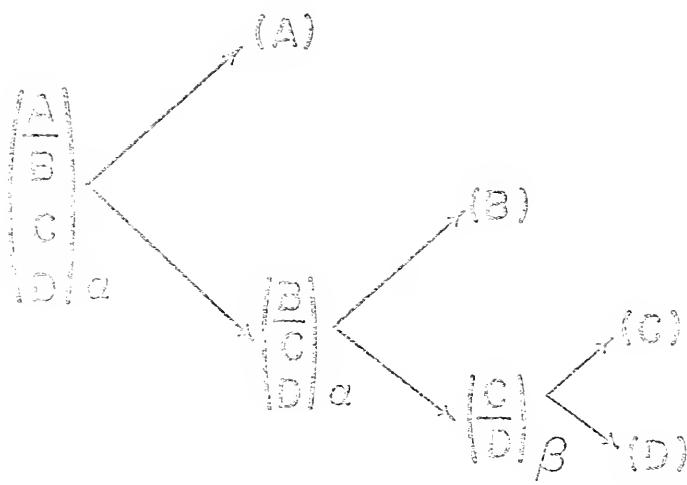


Figure 11.(continued)

VI.3. Branch and Bound Strategy

The goal of the branch and bound strategy is to get the optimum separation scheme by enumerating as few separators as possible.

Hendry and Hughes (1972), with the Dynamic Programming scheme they employed, have to search over the whole set of distinct separators.

The branch and bound strategy of the present method is based on the bounding properties of the dual and the primal return function evaluations. These bounding characteristics were used by McCalliard and Westerberg (1972) to determine whether a feasible modification to a given feasible structure will appreciate system return without having to optimize fully the modified system. In the present work the primal and dual values are used in connection with a branch and bound scheme to restrict the search space of distinct separators and to yield eventually the optimum structure. It is a step by step building procedure of the optimum flowsheet. Since the two-level optimization method (Lasdon, 1964) is the theoretical foundation of the proposed strategy, let us describe this method by considering the following sequential problem (for more details on the two-level method see Chapter II):

$$\text{Minimize } \phi = \sum_{i=1}^n \phi_i(x_i, u_i) \quad (\text{VI-P1})$$

$$\text{s.t. } x_1 = x_1^0 \quad (\text{given})$$

$$x_{i+1} = f_i(x_i, u_i) \quad i = 1, \dots, N$$

$$(x_i, u_i) \in S_i \quad i = 1, \dots, N$$

The Lagrangian of this problem is

$$\begin{aligned} L &= \sum_{i=1}^n \{\phi_i - \lambda_i^T x_i + \lambda_i^T + 1 f_i\} \\ &= \sum_{i=1}^n \ell_i(x_i, u_i, \lambda_i, \lambda_{i+1}) + \lambda_1^T x_1^0 \end{aligned} \quad (\text{VI-1})$$

where the natural boundary condition on λ_{n+1} yields $\lambda_{n+1} = 0$.

Next we define the problem (VI-P2) as

$$h(\lambda_1, \dots, \lambda_n) = \sum_{i=1}^n \underset{(x_i, u_i) \in S_i}{\text{Minimize}} \ell_i + \lambda_1^T x_1^0 \quad (\text{VI-P2})$$

and finally the dual problem (VI-P3)

$$\underset{\lambda_1, \dots, \lambda_n}{\text{Maximize}} h(\lambda_1, \dots, \lambda_n) \quad (\text{VI-P3})$$

$$\lambda_1, \dots, \lambda_n$$

The two-level method involves solving (VI-P3) by first guessing the multipliers and then iterating until $h(\lambda_1, \dots, \lambda_n)$ is maximized. The following theorem (Lasdon, 1970) establishes the dual function as a lower bound of the optimum value:

Theorem:

$$h(\lambda_1, \dots, \lambda_n) \leq \phi(x_1, u_1, x_2, u_2, \dots, x_n, u_n)$$

for all (x_i, u_i) $i=1, \dots, N$ satisfying the constraints of problem (VI-P1) and for all $\lambda_i \in D$, $i=1, \dots, N$ where D is the domain of definition of the Lagrange multipliers.

A direct consequence of the above theorem is the following inequality

$$(lower\ bound) \quad h(\lambda_1, \dots, \lambda_n) \leq \psi(x_1^*, u_1^*, \dots, x_n^*, u_n^*)$$

where x_i^*, u_i^* , $i=1, \dots, N$ is the optimum solution of problem (VI-P1).

Furthermore, it is obvious that

$$(upper\ bound) \quad \phi(x_1, u_1, \dots, x_N, u_N) \geq \phi(x_1^*, u_1^*, \dots, x_N^*, u_N^*)$$

where $(x_1, u_1, \dots, x_N, u_N)$ satisfy the constraints of (VI-P1). The

lower bound will be called the dual bound and the upper bound the primal bound.

Construction of the optimum flowsheet begins by defining a basic flowsheet. The basic flowsheet can be any separation train which produces the desired products. If the same separation problem has been solved in the past, then the flowsheet available in the literature can be the basic flowsheet. If it is a new problem then the basic flowsheet can be developed as follows. For given Lagrange multipliers the sublagrangians ℓ_i , as defined in equation (VI-1), are minimized with respect to the input and decision variables for every N -component separator. As the first separator we retain the separator with the smallest value of ℓ_i . In the 4-component example of section VI.1., let $(A/BCD)_\alpha$ be the first separator thus selected. The second separator can be any of the separators 7,8,9 or 10 in Fig. 11. Following the same procedure we choose separator 7, $(B/CD)_\alpha$ as the second separator, with the smallest ℓ_i . In the same way we choose separator 24, $(C/D)_\beta$ as the last separator. Then the basic flowsheet is given in Figure 11A. Developing the basic flowsheet we

have used the heuristic "choose the potentially cheapest as the next separator." Thompson and King have proposed essentially this heuristic and its implications when using the infeasible decomposition strategy of the two-level optimization method will be discussed later. The sum $h_b = l_1 + l_7 + l_{24}$ is a dual bound to the optimum value for the basic flowsheet. Next we compute a primal bound for this flowsheet. Let this be ϕ_b^* . Thus if ϕ^* is the minimum cost for the basic flowsheet we have

$$h_b \leq \phi^* \leq \phi_b$$

Once the basic flowsheet and its dual and primal bounds are established, the screening of the other alternate structures goes in the following systematic way. If any of the 4-component separators 2,3,4,5 or 6 has a sublagrangian $l_i > \phi_b$, all the flowsheets starting from this separator are disregarded since they will appreciate a greater cost. Thus we develop a tree, each branch of which proceeds as long as the sum of the sublagrangians of the separators contained in this branch does not exceed the primal bound ϕ_b of the basic flowsheet. In Figures 14 and 15 this strategy is demonstrated in the n-butylene purification example of 6 components using two separation methods, distillation (method α) and extractive distillation (method β).

At the end of the above procedure a small number of flowsheets is retained, all of which have a dual bound lower than ϕ_b . The optimal flowsheet is a member of this family of flowsheets. At this point all the retained flowsheets are further examined by the designer. All of the flowsheets may be considered as candidates for the final process and the choice of one of them may be based on many other factors such

as operational, controllability, etc. If desired, further screening of the flowsheets to produce the optimum proceeds in the following manner. The flowsheet with the lowest dual bound is established as the basic flowsheet and a new primal bound is computed. If the dual bound of any of the remaining flowsheets exceeds the primal bound of the new basic flowsheet, this flowsheet is rejected. By continuously updating the dual bound of the basic flowsheet through a change in the Lagrange multipliers, the gap between the primal and the dual bounds of the current basic flowsheet becomes smaller and the screening procedure more effective. In this manner we can continue until very few and probably only one flowsheet is left which, because of the procedure followed, is the "global" optimum. The mathematical rigorousness of the two-level optimization method guarantees this result.

An additional flexibility can be introduced at the first level of screening by properly and timely changing the basic flowsheet. If for example an attractive alternate structure is established during the first screening, a new basic flowsheet may be established based on this flowsheet. A new primal bound is computed and hopefully is lower than the primal bound of the previous basic flowsheet. Only under this condition would the change in the basic flowsheet be useful. It is obvious that such a variation in the initial strategy is highly heuristic. The decision that a certain structure will appreciate a better primal bound and therefore will be a better basic flowsheet depends largely on the intuition and the experience of the designer. Despite the heuristic nature of such decisions, the mathematical

rigorousness of the overall strategy is not lost, since the actual screening is based on the rigorous upper and lower bounds of the optimum costs.

The set of all the distinct separators can also be reduced with the use of certain heuristics. If a certain split is very easy using distillation, the separator with the same split but using extractive distillation is not considered (Hendry and Hughes, 1972). Furthermore, certain separators are physically infeasible; e.g., the feed to an extractive distillation column is immiscible or the two streams in an extraction unit are miscible. Finally, rejection may result if during the simulation of the separation units, difficulties in the performance of units are encountered such as very large internal flows, very high temperatures (Thompson and King, 1972) etc. In all the above cases the separators are excluded either in advance or during the construction of the flowsheets.

The success of the branch and bound strategy depends largely on the proper selection of the Lagrange multiplier values and the efficient estimation of a primal bound for the basic flowsheet. The problem of the synthesis of a separation sequence has certain characteristics which can make it insensitive to the value of the Lagrange multipliers. If the inputs and the outputs for each unit are considerably constrained into a small range of acceptable values, then whatever the values of the Lagrange multipliers are, the range of possible values for the overall dual bound is very limited. This limitation in values can be understood better by considering the simple two-stage example in Figure 12. The dual bound is $h(\lambda)$ defined by

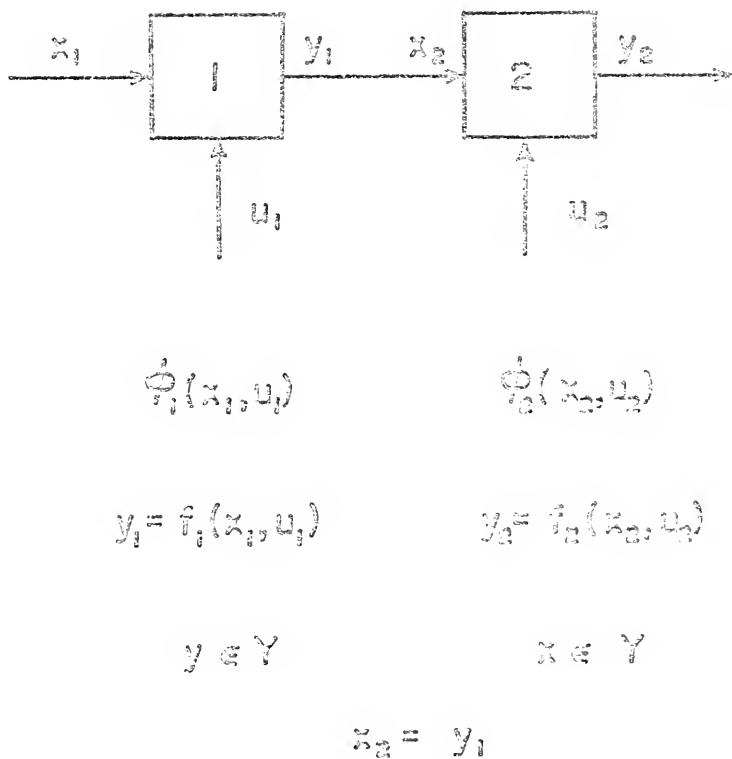


Figure 12. Two-stage Example to Demonstrate the Insensitivity of the Dual Function to the Values of the Lagrange Multipliers

$$\underset{x_1, u_1}{\text{Min}} \quad \{\phi_1(x_1, u_1) + \lambda y_1 \mid y_1 = f_1(x_1, u_1), y_1 \in Y\}$$

$$+ \underset{x_2, u_2}{\text{Min}} \quad \{\phi_2(x_2, u_2) - \lambda x_2 \mid y_2 = f_2(x_2, u_2), x_2 \in Y\}$$

Since both y_1 and x_2 are restricted to a very small allowable set of values, Y , the contribution to the dual bound $\lambda(y_1 - x_2)$ is very small regardless of the value of λ . This characteristic was exploited in this work and the Lagrange multipliers were set equal to zero. A comparison of the values of the λ_i 's with $\lambda=0$ and the values of the objective functions ϕ_i 's for the optimal primal solution did not differ significantly, justifying this approach.

The value of the primal bound of the basic flowsheet is also very important for an effective branch and bound strategy. It must lie as close as possible to the dual bound for an effective screening. This cannot be guaranteed by an arbitrary set of input and control variables for each unit. Thus repeated iterations may be needed to establish a good primal bound. In the case that a similar separation sequence has been synthesized in the past, the optimum cost for that sequence will be a very good primal bound.

VI.4. Examples

In order to illustrate the strategy proposed in the previous section, two separation problems were tackled. These examples were taken from the existing literature in order to allow for comparison and also to provide for a certain realism of the procedure adopted.

VI.4.a. Example 1: n-butylene purification system

This example was taken from Hendry and Hughes (1972) and its purpose is to illustrate the approach developed in the previous sections. The composition of the initial mixture, the flowrates and the destination of each component are given in the work mentioned above and are shown in Table 5. Two separation methods will be considered, distillation and extractive distillation with a furfural-water mixture (4% water by weight). For the simulation of the separation units, the short-cut programs developed by Thompson and King (1972) were used. The thermodynamic routines from the same work were used to calculate the thermodynamic properties of the above hydrocarbon mixture. The objective function to be minimized is the venture cost given by:

$$V = 0.5 * (OP) + 0.25 * (CAP)$$

where

V = venture cost, \$/yr.

OP = operating cost, \$/yr.

CAP = capital cost, \$

TABLE 5

INITIAL FEED TO THE n-BUTYLENE PURIFICATION SYSTEM

<u>Component</u>	<u>Symbol</u>	<u>Flowrate</u>	<u>Mole Fraction</u>	<u>Destination</u>
Propane	C3	10.0	0.0147	Storage
n-Butane	NB	341.0	0.5029	Recycle
Butene-1	B1	100.0	0.1475	Butylene Dehydrogenation
Trans-Butene-2	B2T	106.0	0.1563	Butylene Dehydrogenation
Cis-Butene-2	B2C	81.0	0.1196	Butylene Dehydrogenation
Pentane	C5	40.0	0.0590	Storage

The optimization variables for each unit were the pressure and the reflux ratio factor as the decision variables of a column and the temperature and pressure of the feed in the unit as the input variables.

The ranked lists for the two separation methods are as follows:

Distillation (method α): C3,B1,NB,B2T,B2C,C5

Extractive Distillation (method β): C3,NB,B1,B2T,B2C,C5

The following heuristic was used (Hendry and Hughes, 1972) to reduce the number of possible separators: "Because of the relatively small amounts of propane and pentane present in the mixture and the ease of separation of these components from others by using straight fractionation, extractive distillation splits with either of these components as keys are prohibited." 64 distinct separators can be generated. In Figure 13 the generation of the basic flowsheet is shown, while in Figures 14 and 15 the branch and bound strategy for this example is developed starting with the modes which were not rejected after the basic flowsheet was established. The numbers in a circle in Figures 13, 14 and 15 correspond to the distinct separators shown in Figures 16a, 16b and 16c.

At the end four flowsheets (including the basic) shown in Figure 17 were retained: From a total of 64 distinct separators only 43 had to be optimized. Further screening shows that flowsheet (III), Figure 17, is the optimum with a cost 861,400 \$/yr. This is the same solution found by Hendry and Hughes. We should point out that separate optimization of the flowsheets retained yields the results shown in Table 6. We can notice the relatively small differences in the cost among the four retained flowsheets. Thus for engineering purposes we could have accepted any of the 4 as the optimum or nearly optimum flowsheet.

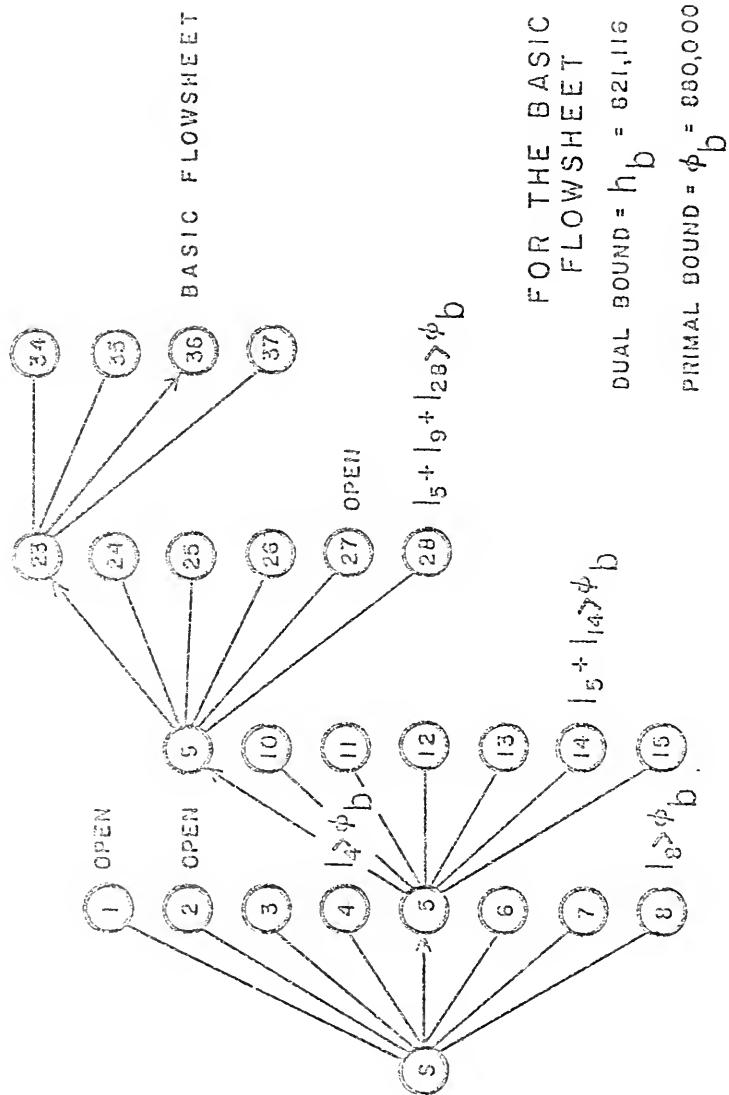


Figure 13. Generation of the Basic Flowsheet for the Synthesis of the n-Butylene Purification System

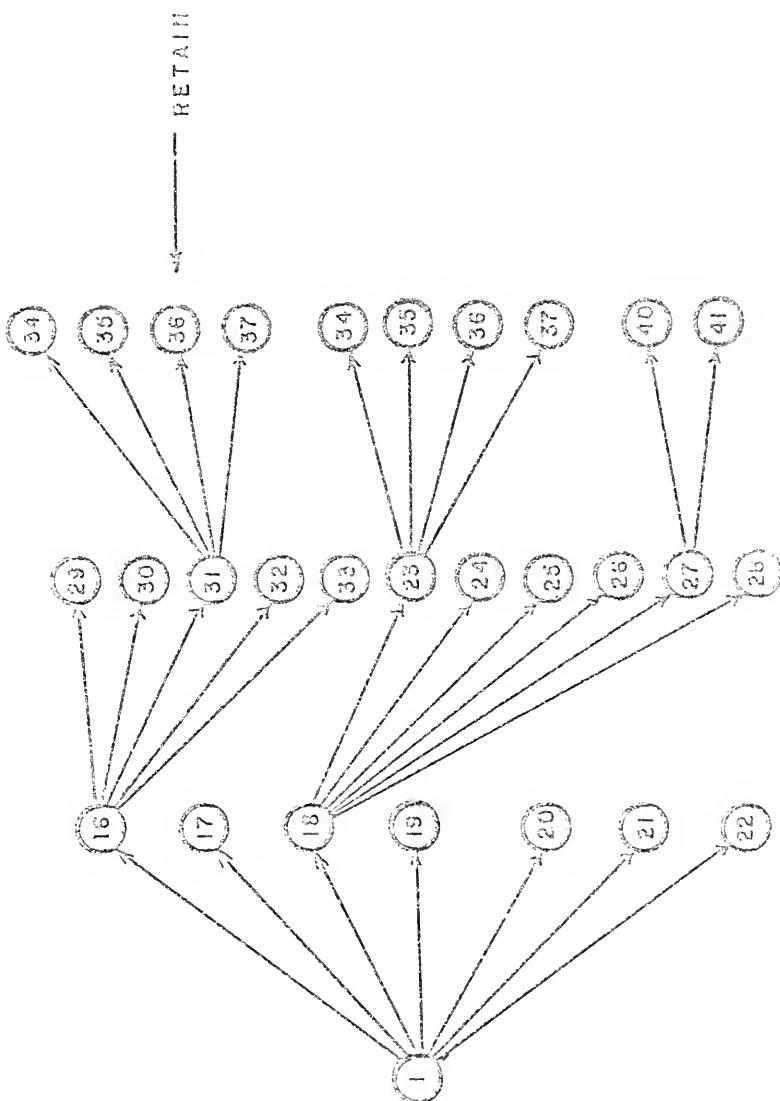


Figure 14. Generation of all the Flowsheets Starting with Separator 1 for the n-Butylene Purification System Example 1

Figure 15. Generation of all the Flowsheets Starting
With Separators 2 or 5 for the n-Butylene
Purification System Example

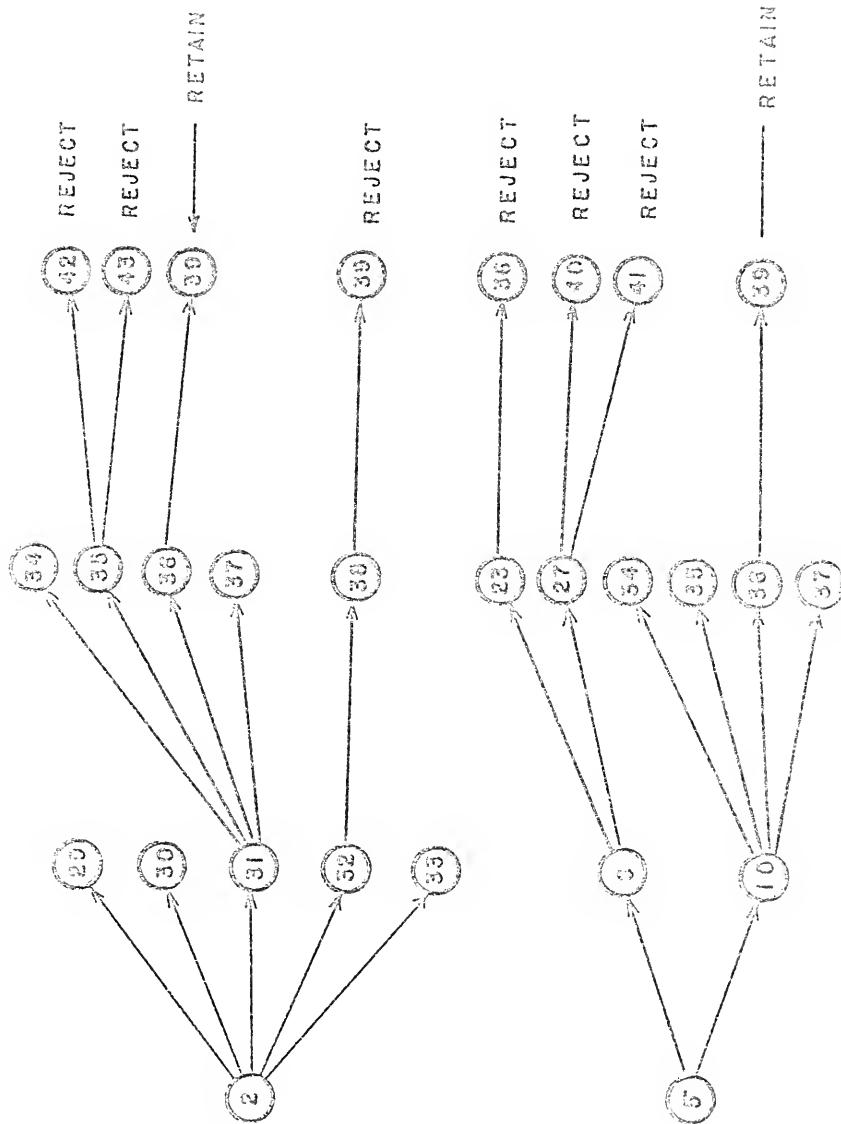


Figure 16. All the Distinct Separators Employed During the Synthesis by Branch and Bound of the n-Butylene Purification System

S - COMPONENT SEPARATORS

1.	$\frac{C_3}{B_1}$	2.	$C_3 \left(\frac{B_1}{B_2} \right)$	3.	$C_3 \left(\frac{B_1}{N_B} \right)$	4.	$C_3 \left(\frac{B_1}{B_2 T} \right)$	5.	$C_3 \left(\frac{B_1}{B_2 C} \right)$	6.	$C_3 \left(\frac{B_1}{C_5} \right)$	7.	$C_3 \left(\frac{B_1}{N_B} \right)$	8.	$C_3 \left(\frac{B_1}{B_2 T} \right)$	9.	$C_3 \left(\frac{B_1}{B_2 C} \right)$	10.	$C_3 \left(\frac{B_1}{C_5} \right)$		
$b_1 = 32,520$	$b_2 = 126,534$	$b_3 = 4300,000$	$b_4 = 900,012$	$b_5 = 27,585$	$b_6 = 27,585$	$b_7 = 90,502$	$b_8 = 97,341$	$b_9 = 1,016,411$	$b_{10} = 242,471$	$b_{11} = 377,803$	$b_{12} = 977,803$	$b_{13} = 977,803$	$b_{14} = 977,803$	$b_{15} = 977,803$	$b_{16} = 977,803$	$b_{17} = 2,203,153$	$b_{18} = 856,532$	$b_{19} = 85,300$	$b_{20} = 10,41,932$	$b_{21} = 303,311$	$b_{22} = 1,163,577$

S - COMPONENT SEPARATORS

(A)

9.	$\frac{C_3}{B_1}$	10.	$C_3 \left(\frac{B_1}{N_B} \right)$	11.	$C_3 \left(\frac{B_1}{B_2 T} \right)$	12.	$C_3 \left(\frac{B_1}{B_2 C} \right)$	13.	$C_3 \left(\frac{B_1}{C_5} \right)$	14.	$C_3 \left(\frac{B_1}{N_B} \right)$	15.	$C_3 \left(\frac{B_1}{B_2 T} \right)$	16.	$C_3 \left(\frac{B_1}{B_2 C} \right)$	17.	$C_3 \left(\frac{B_1}{C_5} \right)$		
$b_9 = 27,243$	$b_{10} = 242,471$	$b_{11} = 377,803$	$b_{12} = 977,803$	$b_{13} = 977,803$	$b_{14} = 977,803$	$b_{15} = 977,803$	$b_{16} = 977,803$	$b_{17} = 977,803$	$b_{18} = 977,803$	$b_{19} = 977,803$	$b_{20} = 977,803$	$b_{21} = 977,803$	$b_{22} = 977,803$	$b_{23} = 243,353$	$b_{24} = 243,353$	$b_{25} = 243,353$	$b_{26} = 243,353$	$b_{27} = 243,353$	$b_{28} = 243,353$
17.	$\frac{B_1}{B_2 T}$	18.	$B_1 \left(\frac{B_2 T}{B_2 C} \right)$	19.	$B_1 \left(\frac{B_2 T}{C_5} \right)$	20.	$B_1 \left(\frac{B_2 C}{C_5} \right)$	21.	$B_1 \left(\frac{B_2 C}{B_2 T} \right)$	22.	N_B	23.	$B_1 \left(\frac{B_2 C}{B_2 T} \right)$	24.	$B_1 \left(\frac{B_2 C}{C_5} \right)$	25.	$B_1 \left(\frac{B_2 C}{C_5} \right)$		
$b_{17} = 2,203,153$	$b_{18} = 856,532$	$b_{19} = 85,300$	$b_{20} = 10,41,932$	$b_{21} = 303,311$	$b_{22} = 1,163,577$	$b_{23} = 1,163,577$	$b_{24} = 1,163,577$	$b_{25} = 1,163,577$	$b_{26} = 1,163,577$	$b_{27} = 1,163,577$	$b_{28} = 1,163,577$	$b_{29} = 1,163,577$	$b_{30} = 1,163,577$	$b_{31} = 1,163,577$	$b_{32} = 1,163,577$	$b_{33} = 1,163,577$	$b_{34} = 1,163,577$	$b_{35} = 1,163,577$	$b_{36} = 1,163,577$

4 - COMPONENT SEPARATORS

$23. \frac{BI}{NB} 24. \frac{BI}{NB} 25. \frac{BI}{NB} 26. \frac{NB}{BI}$ $B2T$ $B2C$ α	$30. \frac{NB}{BI} 31. \frac{NB}{BI} 32. \frac{NB}{BI}$ $B2T$ $B2C$ α	$33. \frac{NB}{BI} 34. \frac{NB}{BI}$ $B2T$ $B2C$ α
$1_{23} = 245,741$ $1_{24} = 3,712,179$ $1_{25} = 342,141$ $1_{26} = 975,100$ $1_{30} = 75,760$ $1_{31} = 762,265$ $1_{32} = 40,633,945$ $1_{33} = 374,355$ $1_{34} = 937,353$	$1_{23} = 245,741$ $1_{24} = 3,712,179$ $1_{25} = 342,141$ $1_{26} = 975,100$ $1_{30} = 75,760$ $1_{31} = 762,265$ $1_{32} = 40,633,945$ $1_{33} = 374,355$ $1_{34} = 937,353$	$1_{23} = 245,741$ $1_{24} = 3,712,179$ $1_{25} = 342,141$ $1_{26} = 975,100$ $1_{30} = 75,760$ $1_{31} = 762,265$ $1_{32} = 40,633,945$ $1_{33} = 374,355$ $1_{34} = 937,353$
(B)	(B)	(B)

Figure 16. (continued)

\mathcal{Z} = COMPONENT SEPARATORS

$$\left(\frac{\partial A}{\partial T}, \frac{\partial B}{\partial T} \right)_{\alpha} \left(\frac{\partial C}{\partial T} \right)_{\alpha} \left(\frac{\partial D}{\partial T}, \frac{\partial E}{\partial T} \right)_{\alpha} \left(\frac{\partial F}{\partial T}, \frac{\partial G}{\partial T} \right)_{\alpha}$$

$$L_A = 3,487,112 \quad L_B = 604,197 \quad L_C = 520,502 \quad L_D = 693,571 \quad L_E = 53,230$$

(C)

\mathcal{Z} = COMPONENT SEPARATORS

$$\left(\frac{\partial A}{\partial T}, \frac{\partial B}{\partial T} \right)_{\alpha} \left(\frac{\partial C}{\partial T}, \frac{\partial D}{\partial T} \right)_{\alpha} \left(\frac{\partial E}{\partial T}, \frac{\partial F}{\partial T} \right)_{\alpha} \left(\frac{\partial G}{\partial T}, \frac{\partial H}{\partial T} \right)_{\alpha} \left(\frac{\partial I}{\partial T}, \frac{\partial J}{\partial T} \right)_{\beta}$$

$$L_A = 14,111 \quad L_B = 19,583 \quad L_C = 274,872 \quad L_D = 543,817 \quad L_E = 521,297$$

Figure 16. (continued)

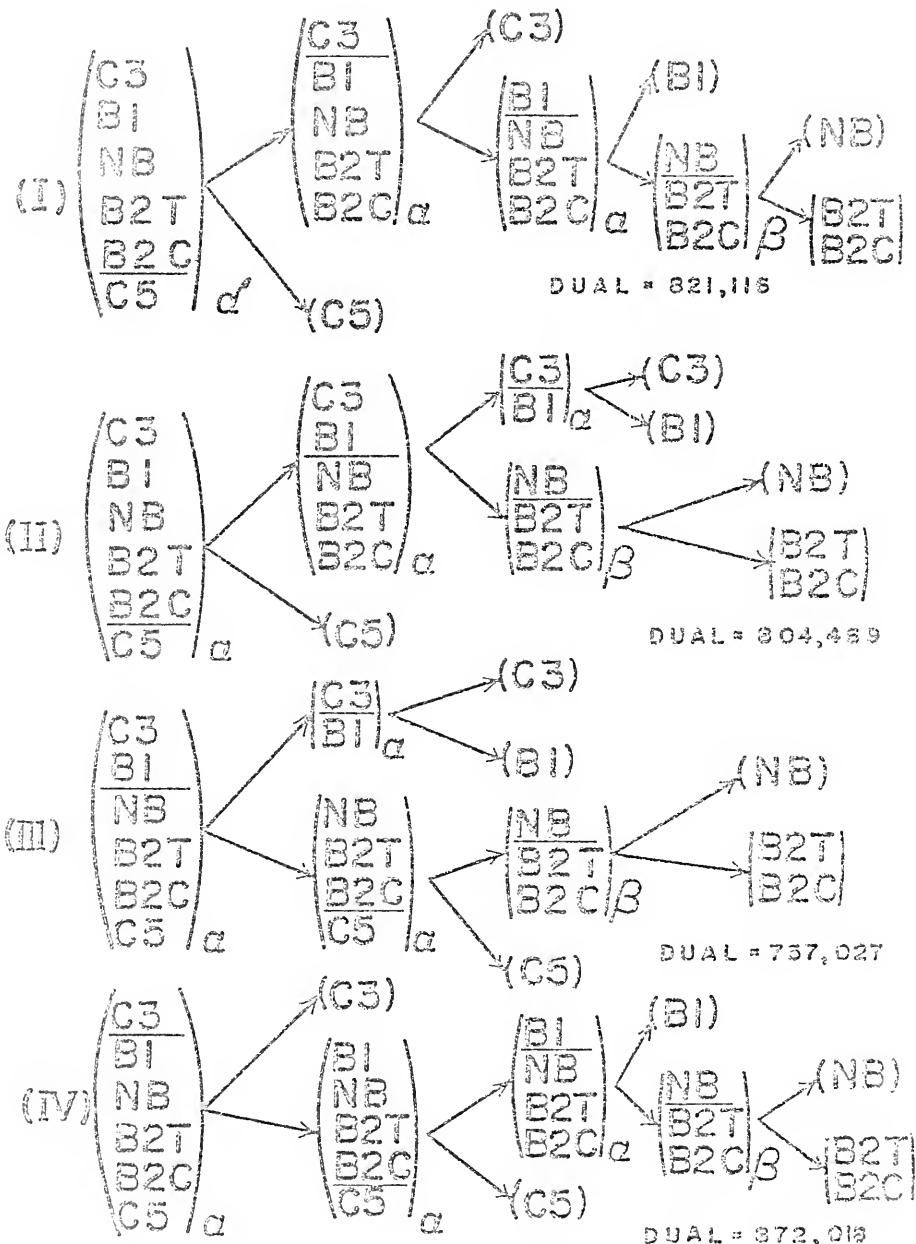


Figure 17. Nearly Optimum Flowsheets Retained at the End of the Branch and Bound Synthesis of the n-Butylene Purification System

TABLE 6
GENERATED FLOWSHEETS AND THEIR MINIMUM COSTS

<u>Flowsheet</u>	<u>Optimum Cost</u>
I	884,828
II	877,512
III	861,400
IV	869,476

VI.4.b. Example 2: Olefins, paraffins separation system

This example was taken from Thompson and King (1972) and is a separation problem of olefins and paraffins into pure products. The specifications of the initial feed and the desired products are shown in the Table 7. Two separation methods were considered: distillation (method α) and extractive distillation with tetrahydrofuran (method β). For the calculation of the thermodynamic properties and the simulation of the distillation and the extractive distillation units the same routines as in the previous example were used. The objective function remained the same, i.e., the venture cost and the optimization variables were as in the previous example. The ranked lists for the two methods are as follows:

RL (method α) : C2,P1,C3,B1,C4,C5

RL (method β) : C2,C3,P1,C4,B1,C5

Applying the same procedure of the branch and bound strategy, 70 distinct separators were generated. Only 34 had to be optimized, see Figures 18a, 18b and 18c. Figure 19 shows the generation of the basic flowsheet which is the flowsheet (V), see Figure 20.

The bounds on this flowsheet are:

dual bound = 602,760 \$/year

primal bound = 791,804 \$/year

At the end the flowsheets shown in Figure 20 were retained (beside the basic). Figures 21, 22, 23 show the flowsheets generated (retained and rejected) during the branch and bound strategy from the open nodes 1, 2, 3 and 4 after after the basic flowsheet was established.

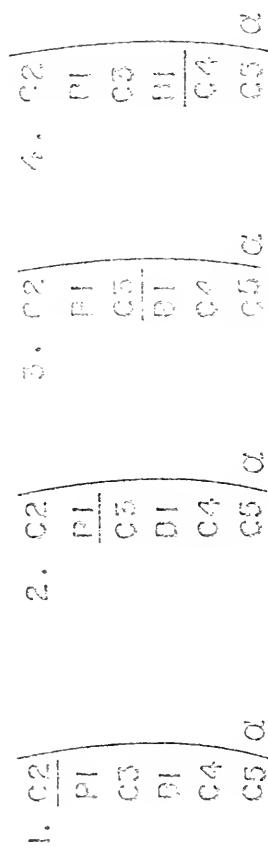
TABLE 7
SPECIFICATIONS OF THE INITIAL FEED
AND THE DESIRED PRODUCTS FOR EXAMPLE 2

Feed rate = 100 lb moles/hr.

<u>Component</u>	<u>Symbol</u>	<u>Feed Mole Fraction</u>	<u>Split Fraction</u>
Ethane	C2	0.20	0.99
Propane	C3	0.20	0.99
Butane	C4	0.15	0.99
Pentane	C5	0.15	0.99
Propene	P1	0.15	0.99
1-Butene	B1	0.15	0.99

Products: 1. Ethane 2. Propane
 3. Butane, Pentane 4. Propene, 1-Butene

S - COMPONENT SEPARATORS



(A)

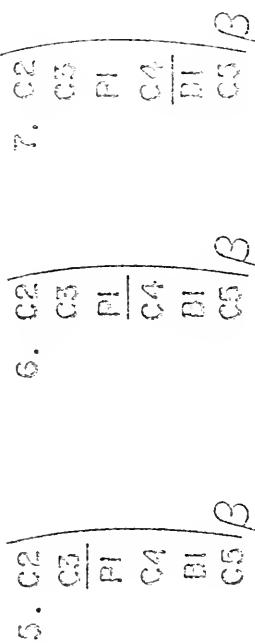


Figure 18. All the Distinct Separators Employed During the Synthesis by Branch and Bound of the Olefins, Paraffins Separation System

Figure 18. (continued)

5 - COMPONENT SEPARATORS

$$3. \quad \left(\begin{array}{c} p_1 \\ c_3 \\ \hline b_1 \\ c_4 \\ c_5 \end{array} \right) \alpha \quad 9. \quad \left(\begin{array}{c} p_1 \\ c_3 \\ \hline b_1 \\ c_4 \\ c_5 \end{array} \right) \alpha \quad 10. \quad \left(\begin{array}{c} p_1 \\ c_3 \\ \hline b_1 \\ c_4 \\ c_5 \end{array} \right) \alpha \quad 11. \quad \left(\begin{array}{c} c_3 \\ p_1 \\ c_4 \\ \hline b_1 \\ c_5 \end{array} \right) \alpha$$

$$|_{18}=328,174 \quad |_{19}=63,351 \quad |_{10}=337,148 \quad |_{11}=4,215,644$$

4 - COMPONENT SEPARATORS

$$14. \quad \left(\begin{array}{c} c_5 \\ b_1 \\ c_4 \\ c_5 \end{array} \right) \alpha \quad 15. \quad \left(\begin{array}{c} c_3 \\ b_1 \\ c_4 \\ c_5 \end{array} \right) \alpha \quad 16. \quad \left(\begin{array}{c} c_2 \\ p_1 \\ c_3 \\ b_1 \end{array} \right) \alpha \quad 17. \quad \left(\begin{array}{c} c_2 \\ p_1 \\ c_3 \\ b_1 \end{array} \right) \alpha$$

$$|_{14}=60,117 \quad |_{15}=342,161 \quad |_{16}=25,609 \quad |_{17}=339,720$$

$$19. \quad \left(\begin{array}{c} c_2 \\ c_3 \\ p_1 \\ c_4 \end{array} \right) \beta \quad 20. \quad \left(\begin{array}{c} c_2 \\ c_3 \\ p_1 \\ c_4 \end{array} \right) \beta \quad 21. \quad \left(\begin{array}{c} c_2 \\ c_3 \\ p_1 \\ c_4 \end{array} \right) \beta$$

$$|_{19}=1,40,751 \quad |_{19}=71,058 \quad |_{19}=1,207,572$$

3 - COMPONENT SEPARATORS

$$22. \left(\frac{C_2}{P_1} \right) \alpha \quad 23. \left(\frac{C_2}{P_1} \right) \beta \quad 24. \left(\frac{P_1}{C_4} \right) \alpha \quad 25. \left(\frac{P_1}{C_3} \right) \alpha \quad 26. \left(\frac{C_2}{P_1} \right) \beta \quad 27. \left(\frac{C_4}{C_5} \right) \beta$$

$$|_{22}=125,973 \quad |_{23}=261,300 \quad |_{24}=210,573 \quad |_{25}=230,530 \quad |_{26}=1,042,751 \quad |_{27}=1,007,210$$

$$28. \left(\frac{C_4}{P_1} \right) \beta \quad 29. \left(\frac{C_2}{P_1} \right) \beta \quad 30. \left(\frac{P_1}{C_3} \right) \alpha$$

$$|_{28}=1173,818 \quad |_{29}=821,042 \quad |_{30}=82,104$$

(C)

2 - COMPONENT SEPARATING

$$31. \left(\frac{C_3}{P_1} \right) \alpha \quad 32. \left(\frac{C_3}{P_1} \right) \beta \quad 33. \left(\frac{C_2}{P_1} \right) \alpha \quad 34. \left(\frac{P_1}{C_3} \right) \alpha \quad |_{31}=40,703 \quad |_{32}=210,573 \quad |_{33}=104,724 \quad |_{34}=203,031$$

Figure 18. (continued)

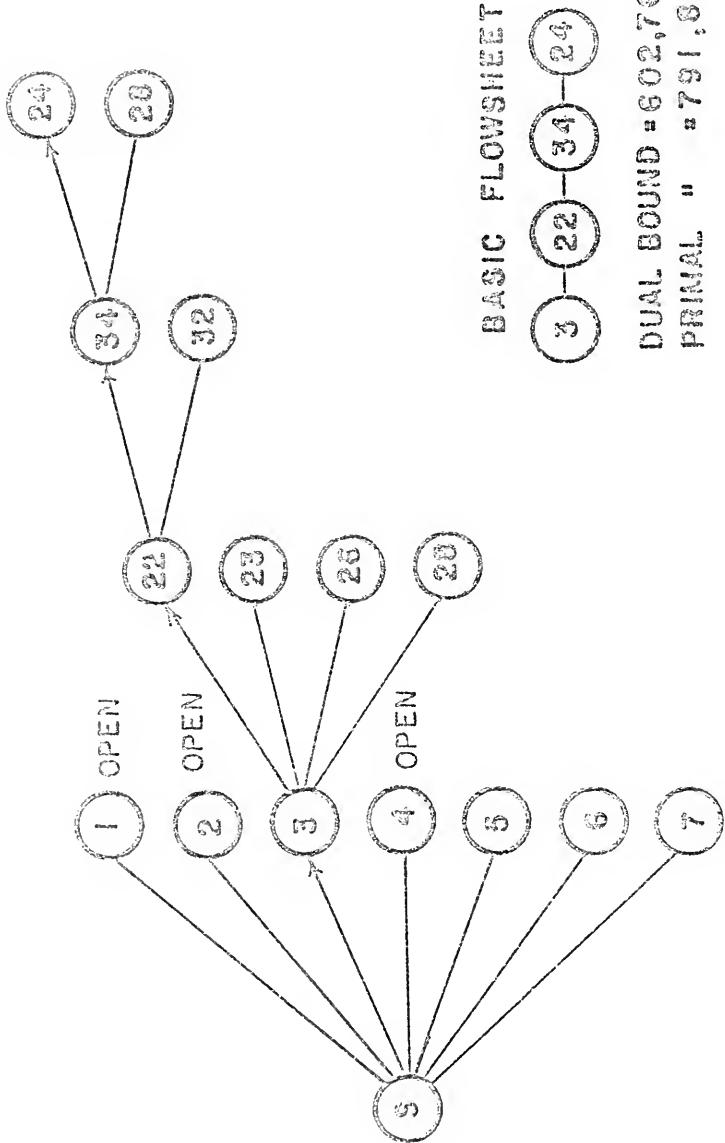


Figure 19. Generation of the Basic Flowsheet for the Synthesis of the Olefins, Paraffins Separation System

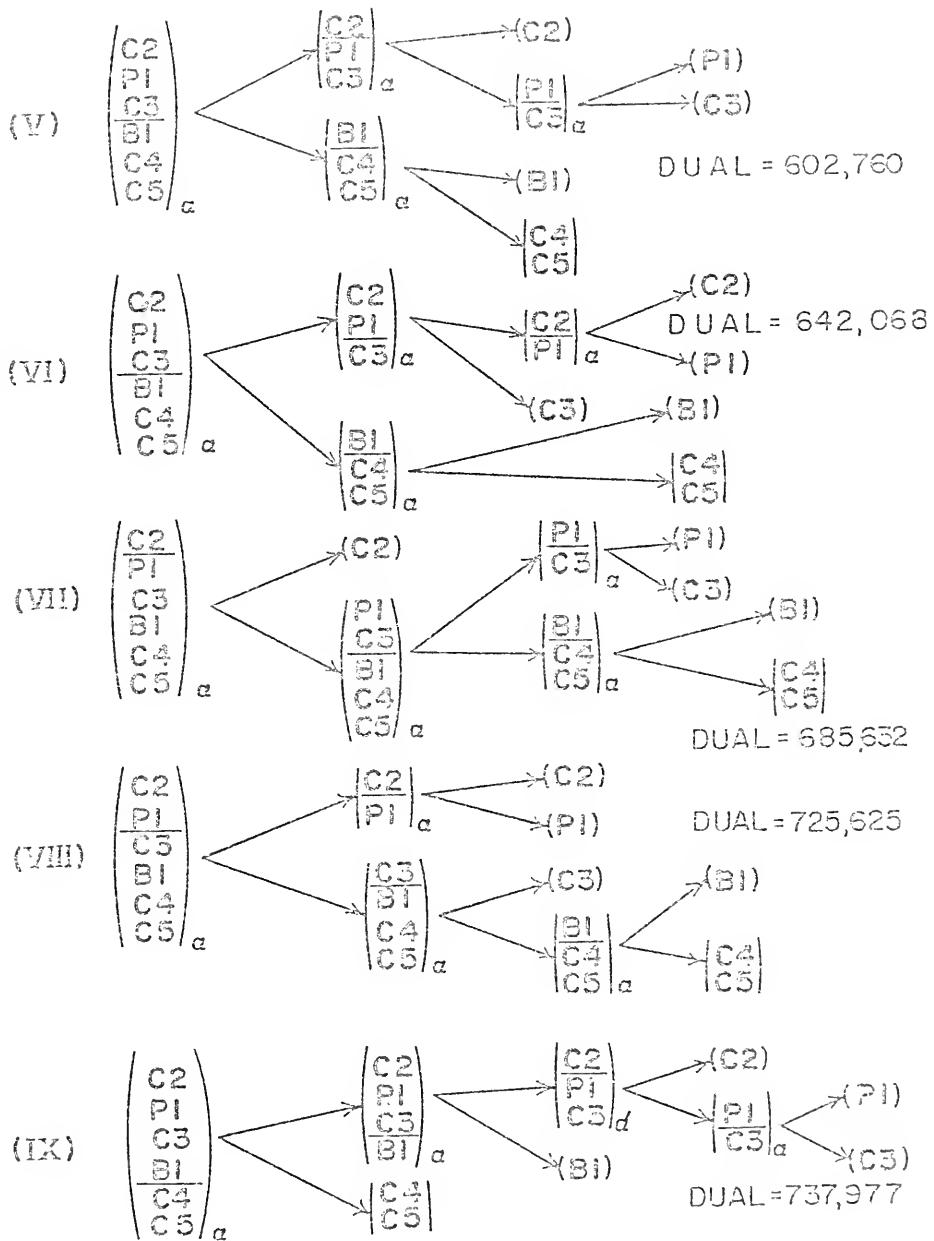


Figure 20. Nearly Optimum Flowsheets Retained at the End of the Branch and Bound Synthesis of the Olefins, Paraffins Separation System

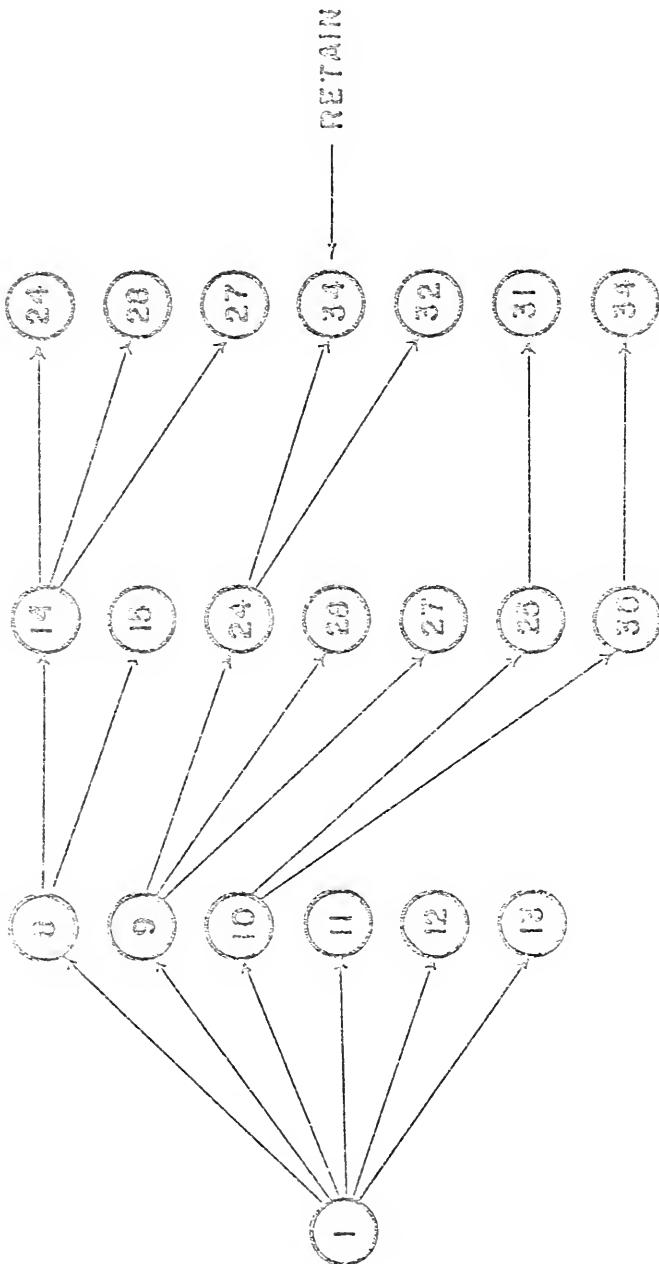


Figure 21. Generation of all Flowsheets Starting with
Separator 1 for the Olefins, Paraffins Separation
System

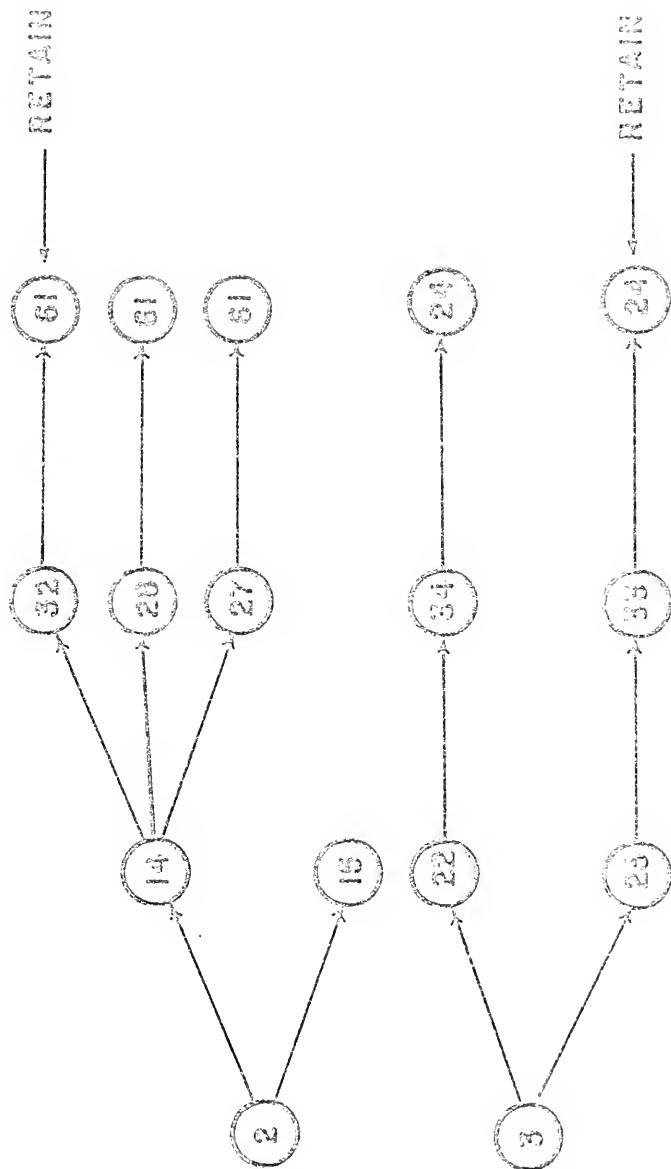


Figure 22. Generation of all Flowsheets Starting With Separators 2 or 3 for the Olefins, Paraffins Separation System

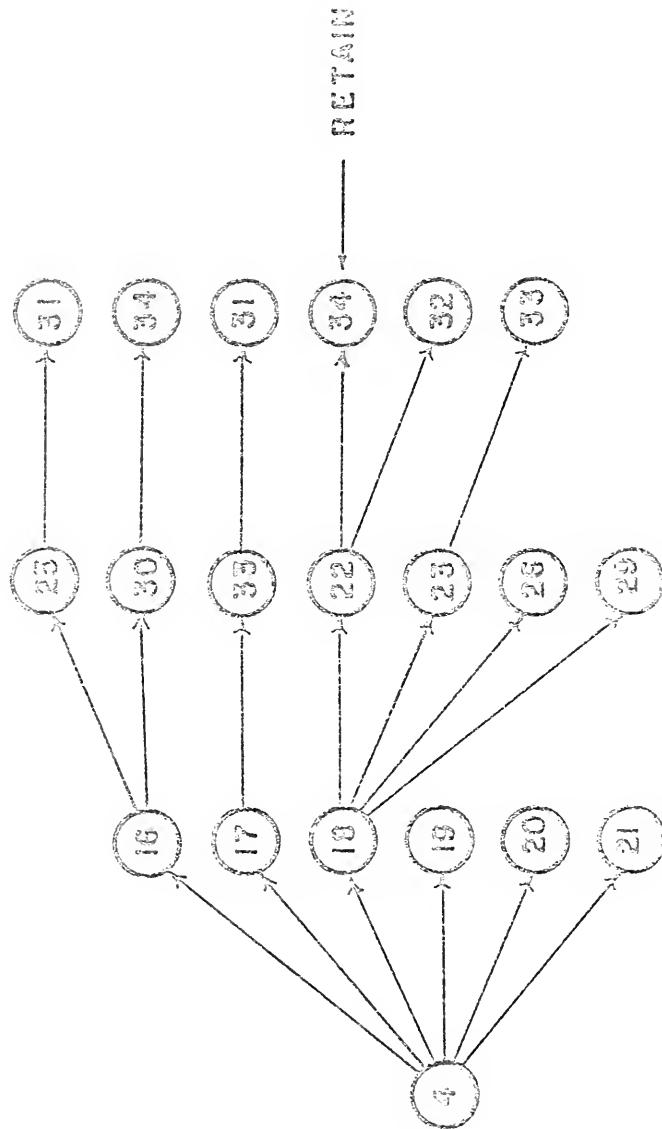


Figure 23. Generation of all Flowsheets Starting with
Separator 4 for the Olefins, Paraffins
Separation System

All the retained flowsheets are nearly optimum. Further screening yielded flowsheet (V) as the optimum one.

VI.5. Discussion

Use of the primal and dual bounds in a branch and bound strategy along with a list processing representation of the separation processes leads to an efficient algorithmic procedure for the synthesis of a small number of nearly optimum multicomponent separation sequences or furthermore for the synthesis of the "globally optimum" separation sequence. The algorithmic nature of the procedure guarantees the optimality of the synthesized sequence unlike the heuristic methods (Thompson and King, 1972; Siirola, et al., 1971). No major assumptions concerning the serial nature of the separation operations are required, as is the case with dynamic programming. For an extractive distillation unit, the removal of the solvent does not have to take place in the next step and the recycling of certain streams in a separation sequence can be easily handled. Thus, mass separating agents can be handled as normal products. In Examples 1 and 2, the composition of the feed to a separator has been considered known, which implies that perfect separation has been assumed. This restriction is easily relaxed. Since the products must follow certain purity specifications, the range of compositions each component can have in the feed mixture to a unit is very limited. Thus relaxing the restriction of specified compositions for the feed in each unit does not appreciably increase the difficulty of the optimization of each separator.

The computation time required by the new procedure is reduced compared to the time required by a dynamic programming scheme, which requires a complete search over the whole set of the distinct separators. (For example 1 only 43 separators from 64 possible were optimized to produce 4 nearly optimum flowsheets, while for the Example 2, 36 separators from 70 possible were optimized to generate 5 flowsheets.)

The basic characteristic of the proposed approach is the capability to produce alternate "nearly optimum" separation sequences quickly and by examining only a subset of the set of all the possible separators. As Example 1 illustrated all the generated sequences differ very little in their minimum cost. Any one of them is almost as good as any other, and the selection of the most suitable sequence can be based on some other factors that the designer will consider important besides the preliminary economics of the unit. At the same time the procedure can, if desired, develop the economically best sequence by screening the final set of the retained flowsheets.

Thompson and King's (1972) proposed strategy consisted of a set of decisions in a sequence and at different stages. Thus first a decision was made for the first separator and then for the second separator and so forth. Despite the iterative character of the procedure, it never assesses the implications of a decision on the subsequent decisions. This characteristic was of course recognized by them and defended because of the desired speed for their method. Unlike this procedure the branch and bound strategy proposed in this work considers effectively the consequences of a decision on the following decisions so that the final process is an optimal entity.

CHAPTER VII

EVOLUTIONARY SYNTHESIS OF PROCESS FLOWSHEETS

In this chapter we try to systematize the logic of evolutionary synthesis. The evolutionary procedure is presented as four subtasks: finding an initial structure, creating a set of rules on how to alter structures, developing a strategy to apply these rules, and finally providing a means to compare structures which are generated. Thus, a sequence of structural modifications based on certain evolutionary rules will ultimately lead to a locally optimum flowsheet. The advantages and disadvantages of the approach will be pointed out and finally an illustration of all the ideas will be made on the synthesis of separation sequences.

VII.1. A General Philosophy on Evolutionary Synthesis

We shall now consider in some detail the synthesis of optimal structures by evolving from less optimal ones through small modifications. Given this goal one must partition the problem into a set of subproblems for which one can develop solutions. We have partitioned the evolutionary synthesis task into the following four subtasks which will be discussed in more detail.

1. To evolve to improved structures, one must have an initial structure which satisfies the design goals.
2. Given an initial structure one needs a set of rules to modify a structure, creating what we shall term its "neighboring" structures.

3. Having the rules to generate modifications, the efficiency of the method will hinge on the strategy used to apply them to evolve to improved structures.

4. Finally one needs a means by which to compare structures to decide if one represents an improvement over an other or not. We shall now try to deal with these subtasks directly and propose certain guidelines which can be used to satisfy them.

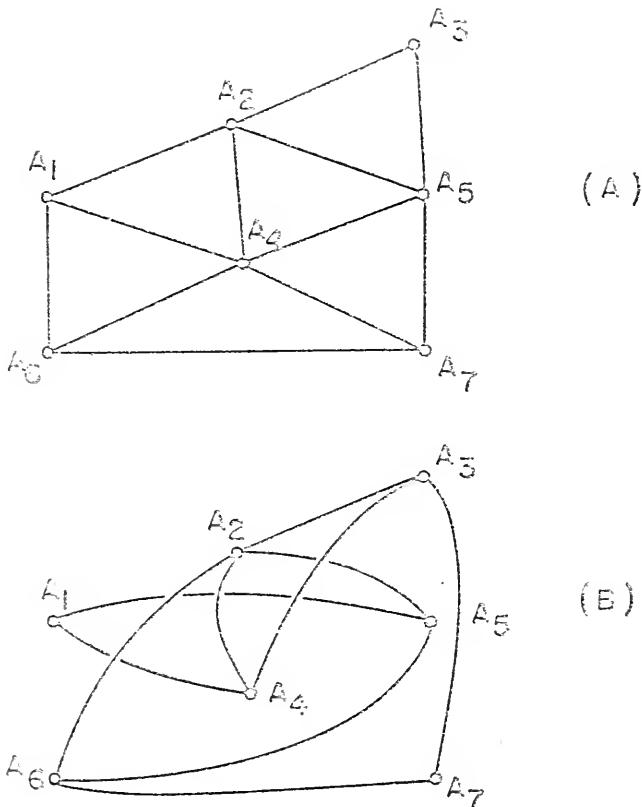
The initial structure. Attempting to locate an optimal structure by evolving to it bears many similarities to optimizing a function of continuous variables. The initial structure corresponds to the initial point and by using our experience in the latter problem, obviously can have a profound effect on the success and efficiency of the method. The better the initial flowsheet, the closer we are to the optimum solution and therefore the faster we attain it. But the development of a good initial flowsheet is not a trivial problem. In fact many of the proposed synthesis strategies deal with this problem. The AIDES system (Siirola and Rudd, 1971; Siirola, Powers and Rudd, 1971) for example develops alternate good initial flowsheets.

Two general approaches exist which can be used to determine a good starting flowsheet: the algorithmic and the heuristic. The first is mathematically rigorous for the conditions stated for the problem and guarantees to produce very reliable alternatives. The branch and bound strategy of the Chapter VI can be viewed as an algorithmic procedure to generate good initial flowsheets. In this case it guarantees that the optimum is a member of the family of the flowsheets generated while any of the other members of this family is

nearly optimum. The algorithmic approach requires extensive computation times. The second, the heuristic approach, does not guarantee any result in a mathematically rigorous manner but tends to produce quickly very good initial flowsheets. Based on rules of thumb which are a result of past experience, ingenuity, and intuition on the part of the designer, they are effective and are likely the more efficient method to develop good starting flowsheets for the evolutionary approach. In case that the same process has been synthesized in the past, the available literature can provide a suitable flowsheet to start the evolutionary strategy. This is because existing plants, although they are not optimal, are structured and operate with a high degree of effectiveness and efficiency. The generation of the initial flowsheet resembles the opening part of the game chess, where the movements are guided by experience and have been well studied.

Once a starting flowsheet is available, the evolution to better flowsheets can begin. Rules are needed to guide the evolution in two different ways: first to develop the possible structural changes that are permitted in the current flowsheet and second to guide the evolution to the most promising direction. We will call the first set of rules evolutionary rules while the second set of rules comprises the evolutionary strategy.

The evolutionary rules. Consider a space containing a node for every possible structure which is expected to meet the design goals. Figure 24A shows such a space. Two nodes will be close to each other in this space if they are judged to be very similar, having only a single "small" structural difference between them. Then the evolutionary



NODES: FLOWSHEETS

EDGES: EVOLUTIONARY RULES

Figure 24. An Illustrative Diagram Showing Two Alternate Sets of Evolutionary Rules for a Family of Flowsheets A_i

rules we wish to invent can be viewed as a set of edges joining these nodes and forming a network as in Figure 24A. Thus the basic notion of the neighboring flowsheet is developed. A flowsheet B will be called neighboring flowsheet to a flowsheet A if B has been generated from A with the smallest possible number of meaningful structural modifications to A. Thus around flowsheet A there is a family of flowsheets with the smallest number of structural differences from A; these are the neighboring flowsheets of A. We will view the evolutionary rules as the guidelines needed to develop all the neighboring flowsheets of the current flowsheet. Figures 24A and B show the generation of the neighboring flowsheets with two different sets of evolutionary rules. Figure 25 shows also the generation of close neighbors using simple structural modifications.

With this discussion we are motivated to require the set of evolutionary rules invented have at least the following properties.

1. Efficiency: the rules should invent legitimate candidate flowsheets only and not ones which are infeasible or undesired for other reasons, such as requiring technology we wish to avoid or which we do not have. For the network of nodes representing the feasible flowsheets, this requirement means no edges lead out of a node which fails to connect to another node.
2. Completeness: to guarantee by repeated application, the potential to generate all possible flowsheets for the system under examination. This property says that the network (graph) representing the structures as nodes and the rules as edges must be connected.
3. Reversibility: the rules should satisfy the requirement that if flowsheet A is a neighbor of flowsheet B, then B is a neighbor of A.

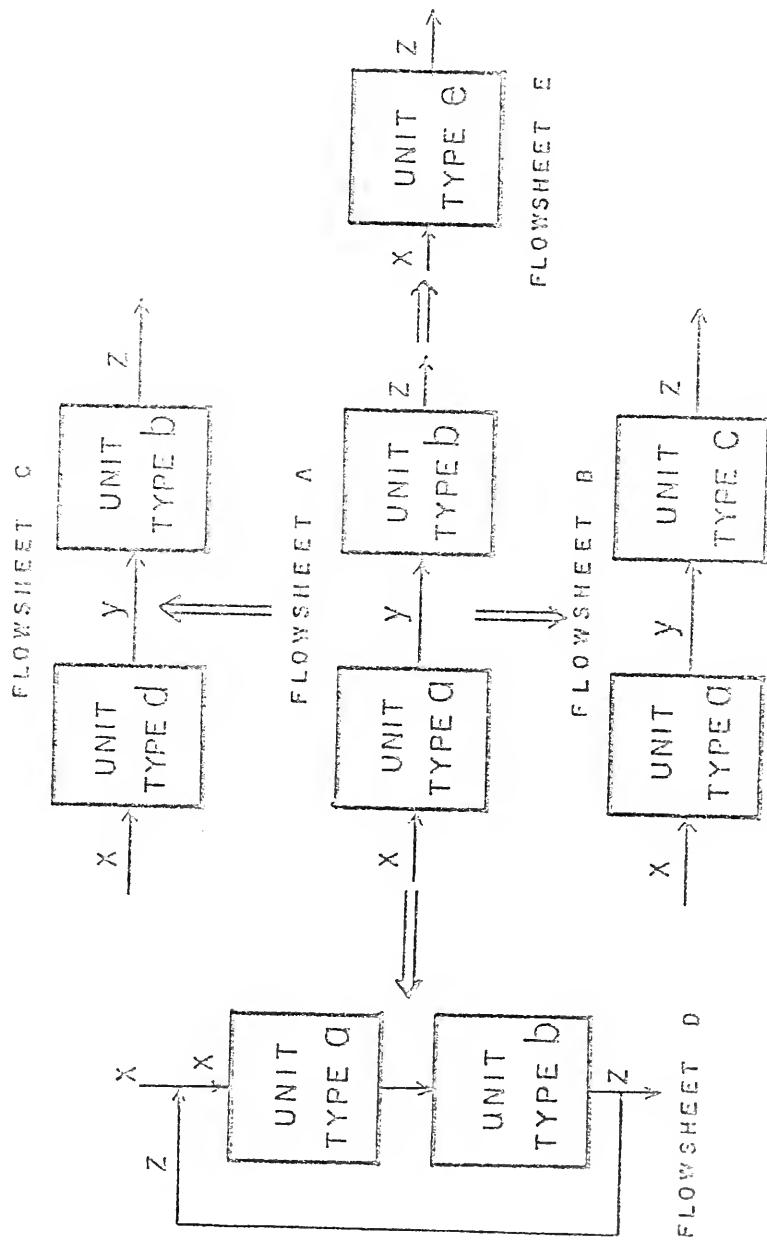


Figure 25. Flowsheet A and Its Neighboring Flowsheets B, C, D and E Resulting from A with Simple Structural Modifications

Again the graph has a corresponding property; that is, the graph should be undirected.

The last property we are inclined to impose cannot be proved but certainly seems desirable.

4. Intuitive reasonableness: the rules should connect structures which a design engineer considers to be actual "neighbors". The rules of the network in the Figure 24A are appealing in this sense whereas the ones in Figure 24B are not.

The first three properties just stated guarantee that, starting from any feasible flowsheet, we can generate in a reasonable manner all possible flowsheets which are feasible and meaningful. Thus by evaluation of all the alternatives one could locate the optimum solution. However, we wish to examine here if a properly selected scheme can lead from the starting flowsheet to the optimum by enumerating only a small subset of the set of all the flowsheets, and the fourth property appeals to one's intuition as a helpful one for this goal.

The evolutionary strategy. The evolutionary strategy is one that will hopefully lead the designer to the optimum flowsheet by an effective use of the evolutionary rules. As such it must be fast and yield good, if not optimum, results. There are different variations that an evolutionary strategy can follow depending largely on the search time that can be spent and the quality of the solution that is desired. The general tendency is to move from a flowsheet to one of its neighboring flowsheets and in particular to the more promising one. Several approaches suggest themselves.

1. One might generate, using the evolutionary rules, all neighboring flowsheets to the current flowsheet, size and cost them rigorously.

Among the alternatives, the one effecting the greatest improvement would be chosen as the next current flowsheet.

2. Another approach is to generate all the neighboring flowsheets but to use heuristics to choose the most likely best next flowsheet. This flowsheet is evaluated and, if it represents an improvement, is chosen as the next current flowsheet. The other flowsheets would not be evaluated. If it fails, the second most likely flowsheet is investigated and so forth. Quite good heuristics are usually available for chemical process design and this approach could prove very effective.

3. The evolutionary rules could be applied selectively, applying only some until a locally optimum structure is found using only them. Then the other rules could be applied to see if they lead to improvements. This approach is rather like a univariate search for continuous variable optimization.

4. If the total number of flowsheets is small or perhaps only when investigating a structure which appears to be locally optimum, one could not only evaluate neighbors but also neighbors of the neighbors (second level of neighbors) and so forth. This idea is suggested by the look-ahead strategy in chess where as many look-ahead steps are investigated as time and space permit. Pho and Lapidus (1973) also used this idea to generate networks of heat exchangers as nodes of a tree and search for the optimum.

Evaluation of the Flowsheets. The last subtask into which we partitioned evolutionary synthesis is providing a means to evaluate flowsheets so they may be compared. Again many options are possible, and we shall suggest a few.

1. Evaluate the flowsheet at a reasonable set of values for the decision variables only. This corresponds to a case study approach and is useful if large differences occur in the results.
2. Optimize each structure by adjusting the decision variables to minimize the objective function. Clearly considerable effort is required here, but the fact that neighboring flowsheets are only slightly different may provide good starting conditions for this optimization.
3. Use primal and dual bounding on each structure as suggested by McGalliard and Westerberg (1972). These bounds provide a range within which the optimum value of the objective function must occur for a flowsheet. Nonoverlapping ranges permit easy comparison. The interesting feature here is that a quite good dual bound for a slightly modified structure is often easy to obtain, requiring only that the modified portion of the structure be analyzed.

Again a mixture of approaches is possible. If a case study approach indicates major reduction in the objective function, then it can be used until the screening becomes more difficult. Then more and more precise evaluations can be used. Only the final structure may require a detailed optimized evaluation of it and its neighbors to prove that it is the best locally.

Whichever of the above strategies is adopted, better and better flowsheets are continuously synthesized until a local optimum is found. This same procedure can be repeated from different starting points. Depending on the nature of the problem one or more local solutions may result.

VII.2. Evolutionary Synthesis of Optimal Multicomponent Separation Sequences

In this section the previously developed notions of evolutionary synthesis will be illustrated on the synthesis of multicomponent separation sequences. The problem of synthesizing such a system was formulated in Chapter VI. Representation techniques for separation schemes, the nature of the neighboring flowsheets, the evolutionary rules and evolution strategy will be developed here in the framework of evolutionary synthesis.

VII.2.a. Representation of separation sequences as binary trees

The notion of the trees is not new. It has been used extensively in different areas. Formally we will define a binary tree as a "finite set of nodes which either is empty, or consists of a root and two disjoint binary trees" which will be called the upper and the lower subtrees of the root. A separation sequence can be represented as a binary tree. Every separator receives one feed and produces two product streams which either are final products or have to be treated further. Thus the set of separators in a separation sequence satisfies the definition of the binary trees as given here. Consider the following separation sequence in Figure 26A. The binary tree corresponding to this flowsheet is given as Figure 28B, where α represents a separator using separation method α and β represents a separator using separation method β . The intermediate nodes of a binary tree will represent separation operators, while the terminal nodes represent basic products (to be called basic operands) which are to be produced from the initial mixture. The edges correspond to streams connecting the different separators. The representation of

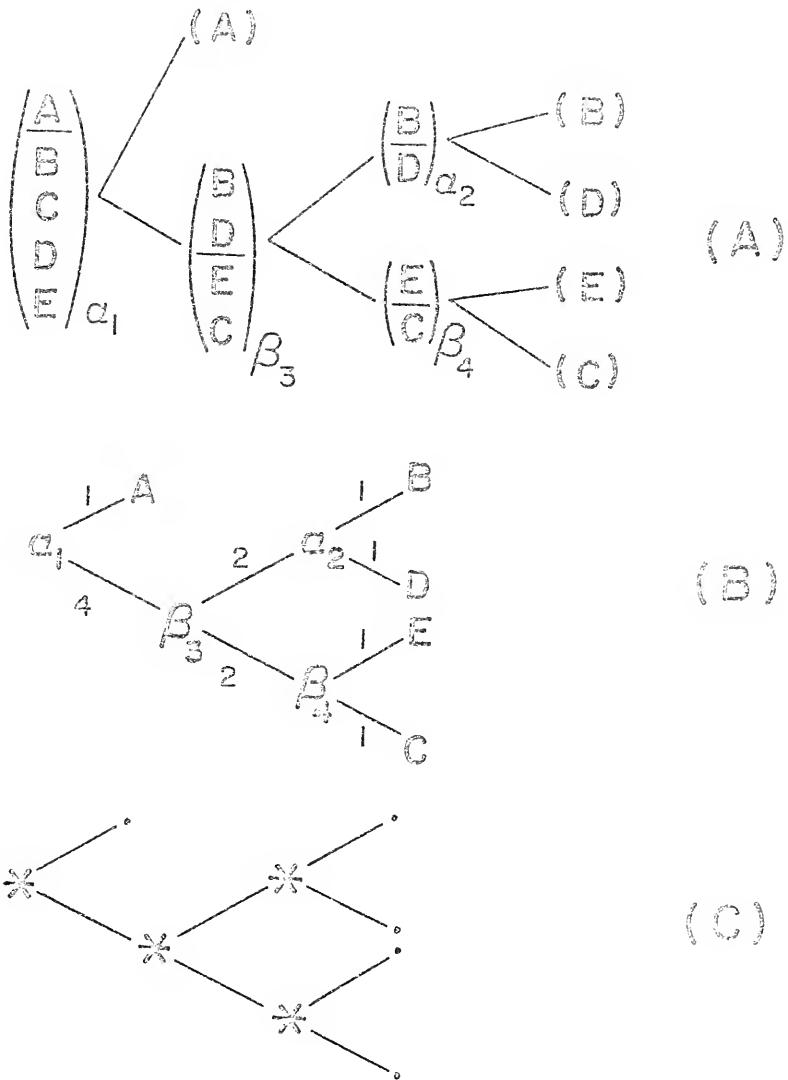


Figure 26. A Separation Sequence (A), Its Corresponding Binary Tree (B) and the Skeleton Structure (C) Corresponding to This Tree

a separation scheme by a binary tree is a compact and computationally very useful approach. Binary trees or their equivalent Polish strings are easily handled in a computer.

If we replace all operators by an asterisk (*) and all basic operands by a dot (.) in a tree, we shall define the resulting tree as the "skeleton structure" equivalent to the original tree. Figure 26C represents the skeleton structure of the tree in Figure 26B. We shall also define the "up operand count" for an operator as the total number of basic operands in the upper subtree emanating from that operator and the "down operand count" as the total number of basic operands in its lower subtree. These counts are given on the tree branches in Figure 26B. For a binary tree representing a separation train the up operand count denotes how many components are in the top product stream for that separator and the down operand count, how many components are in the bottom product stream. Furthermore, if we convert a given skeleton structure back to an actual separation scheme by specifying the kind of separator used in each intermediate node, the positions of the various components of the initial mixture on the binary tree are automatically specified through the use of the ranked lists. For example, if the ranked lists for methods α and β are:

$$RL(\alpha) : ABCDE$$

and

$$RL(\beta) : BDAEC$$

the distribution of the components A, B, C, D and E on the tree in Figure 26B is fixed as shown. Thus, operator α_1 has one up operand

and four down operands. From the ranked list for separation method α these are (A) and (B,C,D,E) respectively. Similarly, for operator β_3 which receives the mixture B,C,D and E, there are 2 up operands and 2 down operands, and from the ranked list for method β these must be (B,D) and (E,C) respectively. In the same way we find the operands for operators α_2 and β_4 .

VII.2.b. Neighboring flowsheets and the evolutionary rules for a separation sequence

Given a certain separation scheme, all the neighboring flowsheets must be generated. For a separation sequence a neighboring flow-sheet must retain as much as possible of the current structure and differ only in a small portion of it (Property 4). Usually this type of change is localized by considering two consecutive separators. Consider the binary tree in Figure 27A. Operator α_4 performs the split ... D/E and operator α_3 the split ... C/D.... We can make a simple modification to the above structure by exchanging the relative positions of operators α_3 and α_4 and therefore the relative position of these two splits in the overall structure. By doing so the structure in Figure 27B results. Comparing these two structures we notice that they differ as a result of one simple structural modification in an isolated part of the overall structure. We shall therefore consider these two flowsheets to be neighbors to each other. We shall refer to structure 27B as a down neighbor to structure 27A because the two separators interchanged are in a "downward" relationship to one another. Property 3 for our evolution rules requires that if structure 27B is a neighbor to 27A, then 27A is a neighbor to 27B. In particular we shall call it an up neighbor.

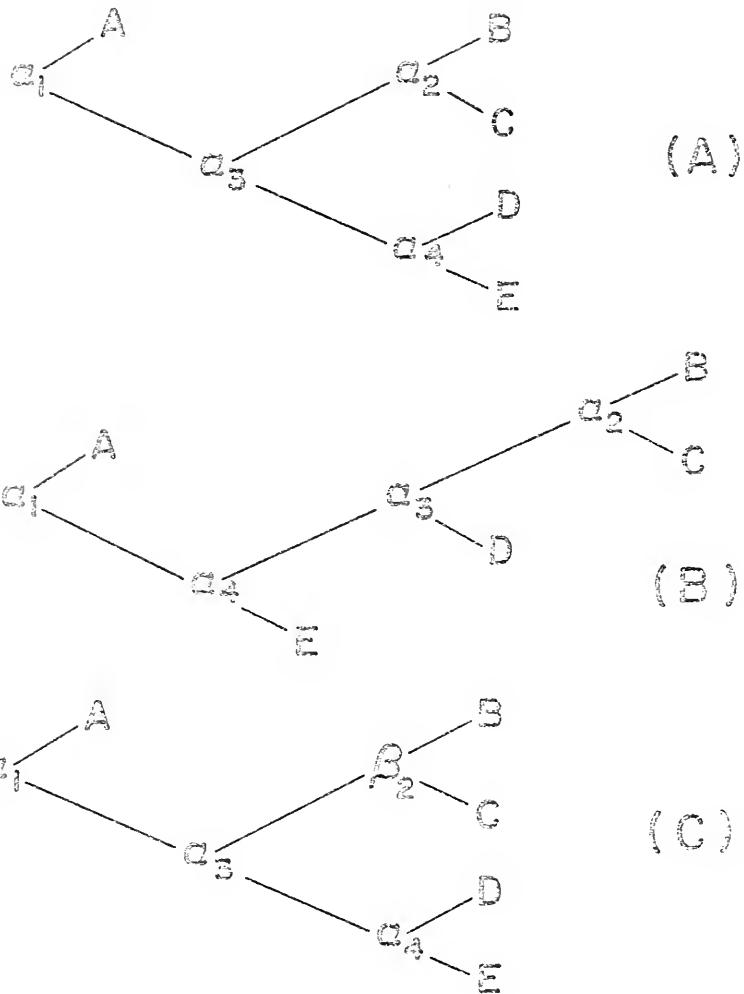


Figure 27. Flowsheet (A), a Down Neighbor (B) to It, and a New Separator Type Neighbor (C) to It

To illustrate further the above relationship between these structures consider the flowsheets they represent, Figures 28A and B respectively. In the first flowsheet, Fig. 28A, the ... D/E split takes place before the ... C/D ... split, while the opposite is the case with the second flowsheet, Fig. 28B. Thus we can say that the difference between them is only in the order which these two splits occur.

A third kind of neighboring flowsheets is generated by substituting an operator with a different kind of operator, for example use separation method β instead of α to attain a certain split. The structure in Figure 27C is a neighboring flowsheet of this type to the one in Figure 27A.

When a tree has more than a single type of separation operator present in it, the interchange (either upward or downward) of two unlike operators may lead to a considerable reshuffling of the basic operands. We shall still use the rules developed for a single operator type to generate neighbors for different operator types as there appears to be no obvious method to make a smaller change to a structure in this case.

Thus it is evident from the above that three evolutionary rules are needed to generate all the neighboring flowsheets to a given flowsheet. These rules are further discussed in the section VII.2.c. from the polish strings point of view.

The evolutionary rules for a separation sequence as discussed above satisfy the first three basic properties required in section VII.1. Thus, Property 1: all the rules create physically consistent and feasible flowsheets since any binary tree resulting from the application

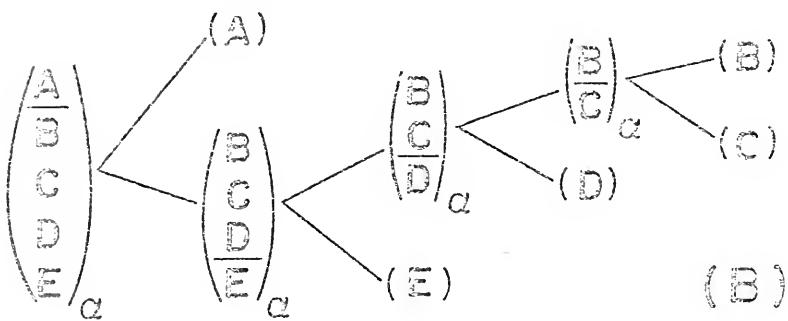
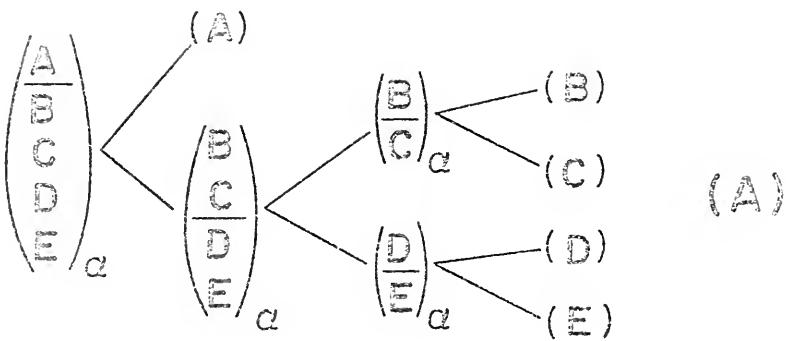


Figure 28. Separation Sequences Corresponding to the Binary Trees of Figures 27 A and B

of the previous rules corresponds to a separation sequence.

Property 2: by repeated application of the evolutionary rules, all possible flowsheets can be generated. The proof of this result is given in the section VII.2.d. This guarantees the completeness of the evolutionary rules. Property 3: if a flowsheet B is an up neighbor of A, then A is a down neighbor of B. These characteristics are also met with the third evolutionary rule, where a separator is replaced by a different kind of separator. The "existence" of property 4 can only be argued and not proved, as we have done in presenting these rules.

VII.2.c. Polish strings and their representation of separation sequences

Binary trees were introduced to represent separation sequences, but for computation purposes polish strings have been used to provide a compact and elegant way to represent binary trees. Polish strings, invented by the Polish logician Lukasiewicz (Knuth, 1968) consist of a string of operators and basic operands. In computer science they are used to represent mathematical expressions and an operator is a basic mathematical operation such as multiply or subtract and a basic operand is an alphabetic or numerical character. For example the algebraic expression

$$3x^2y + z^2/u$$

can be represented by the binary tree shown in Figure 29, which in turn is represented by the following polish string (using FORTRAN operators):

$$+ / u (**) 2 z * y * (**) 2 x 3$$

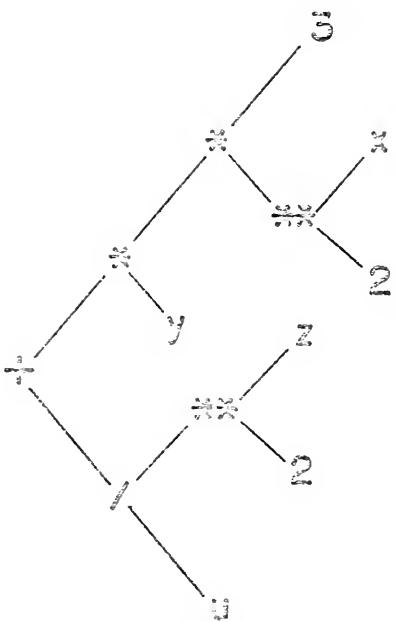


Figure 29. A Binary Tree for the Algebraic Expression
 $3x^2y + z^2/u$

The +, /, **, and * are the operators while the u,z,2,y,x, and 3 are the basic operands. A polish string is interpreted in reverse order (usually referred to as reverse polish) which we can do here to aid in interpreting such a string. First an empty stack for operands is created and if an operand is encountered it is placed on this stack. If an operator is encountered, the operator is immediately applied to the last two operands on the stack. These two operands are removed from the stack and then the result of the operation is placed back on the stack. These steps for the above string are given in Figure 30.

Since a separation sequence is a binary tree, it can be represented by a polish string. In such a polish string an operator is a separator employing one of the available separation methods while a basic operand is a component of the initial mixture. Consider the separation train in Fig. 25B. The polish string representing this flowsheet is:

$$\alpha_1 \beta_3 \beta_4 C E \alpha_2 D B A$$

Since every separation scheme can be represented as a binary tree, there is a polish string for every separation sequence.

Before presenting the evolutionary rules in terms of polish strings let us give some further characterizations. Consider the following string

$$\alpha_1 \alpha_2 \alpha_4 \cdot \alpha_3 \dots$$

In the following string the operands which would be on an operand stack if the string were being interpreted can be counted by starting

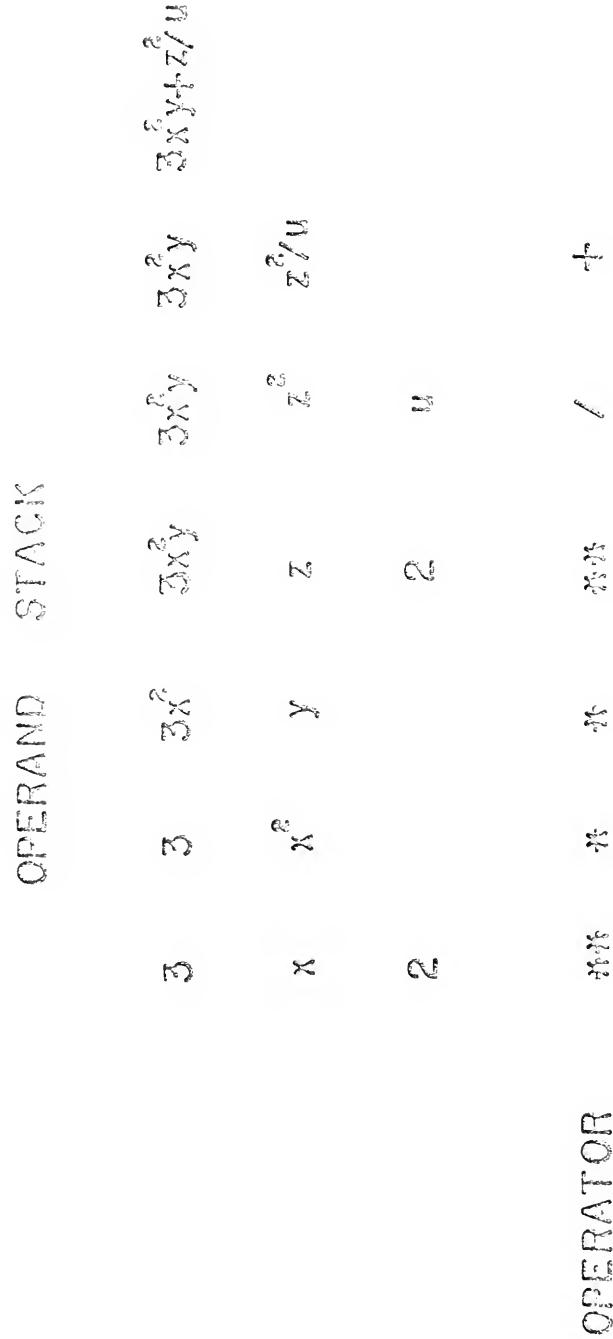


Figure 30. Interpreting a Polish String - Development of an Algebraic Expression Using a Stack of Operands and a Set of Operators

from the right end of the string.

1	2	3	4	3	4	3	2	1
	α_1	α_2	α_4	·	α_3	·	·	·

Note that when an operator is encountered the number of operands decrease by one since an operator employs two operands and produces a single new operand. Following the count of the operands, the two operands of the operator α_1 are readily identified. Thus for operator α_1 we have:

1	2	3	4	3	4	3	2	1	
α_1		α_2	α_4	·	α_3	·	·		·

first operand
second operand

In a similar way the operands of the operators α_2, α_3 and α_4 are identified. The full picture is given in Figure 31.

Let us now see how the evolutionary rules can be used along with a polish string representation in a manner suitable for computer programming.

Rule 1: (Generation of down neighbors). If an operator in a polish string is followed directly by a second operator a down neighbor exists. Move the first operator to a position just before the second operand of the second operator.

Rule 2: (Generation of up neighbors). If an operator is immediately preceded by a basic operand an up neighbor exists. Simply reverse the steps taken in rule 1; the operator being examined here will become the "first" operator of rule 1.

Rule 3: (Generation of neighbors with different kind of separators).

In order to generate all the nieghbors to a given separation train, by introducing a new kind of separator, introduce the new operator (corresponding to the new kind of separator) to every possible position occupied by any other operator in the given string.

In Figure 31 the movement of operators to generate neighboring flowsheets using rules 1 and 2 is also shown, while the flowsheets are shown in Figure 32.

Note to apply a rule requires very simple programming operations. For rule 1, an operator is removed from a string, a number of elements are moved up one position until the second operand of the second operator is inserted back in the string. The simple counting strategy discussed earlier locates directly the second operand of the second operator as the elements are being moved. The reverse steps give rule 2.

VII.2.d. Proof of the completeness of the evolutionary rules

As stated in section VII.1. the evolutionary rules must have the characteristic of completeness, i.e. all possible flowsheets can be reached using repeated applications of rules 1, 2 and 3 from any initial flowsheet.

The proof of the completeness of the evolutionary rules given here for separation systems will be by providing an algorithm which can convert any given original flowsheet to any other target flowsheet. Three steps constitute the heart of the algorithm:

Step I: Convert the original flowsheet to a binary tree structure with the same kind of operators (i.e. α) by at most $n-1$ repeated applications of rule 3 only, where n is the number

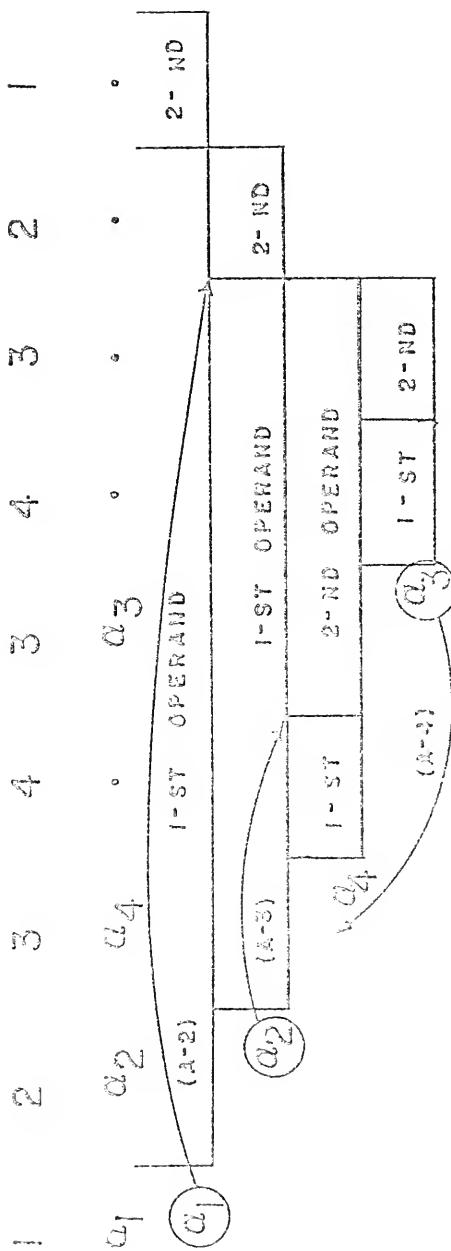
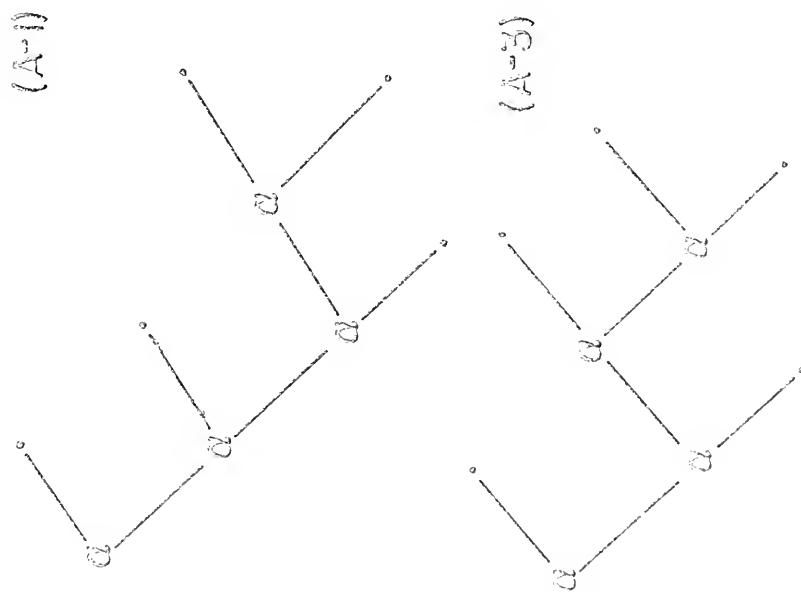
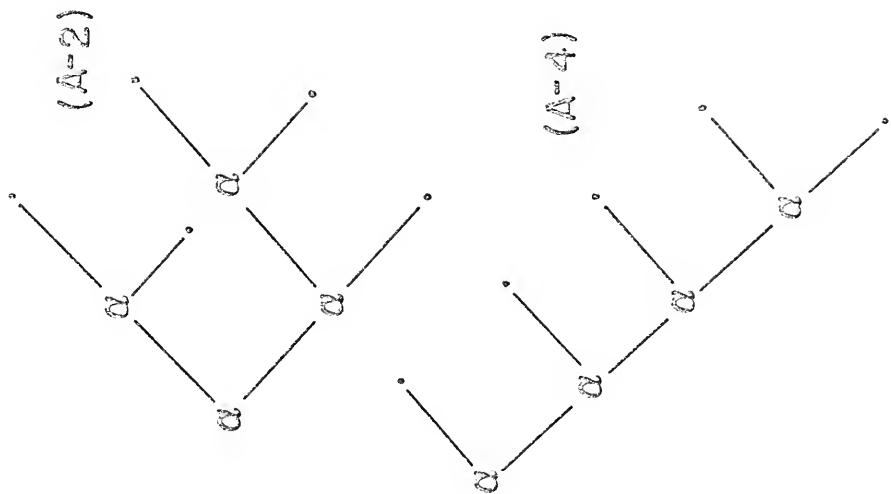


Figure 31. The Operators of a Polish String and Their Operands, and the Schematic Representation of the Generation of the Neighboring Flowsheets by Applying Rules 1 and 2

Figure 32. Flowsheet A-1 and Its Neighboring Flowsheets A-2, A-3 and A-4 Generated Using Rules 1 and 2



of components in the original mixture.

Step II: Change the skeleton structure of this flowsheet to that of target flowsheet by a finite number of applications or rules 1 and 2 only.

Step III: Convert the final binary tree structure to the target flowsheet by at most $n-1$ repeated applications of rule 3 only.

It is clear from the above that only step II requires a proof that it can be accomplished, since the validity of the steps I and III is self-evident.

To prove step II, first list the basic operands according to the ranked list of the separation method used (i.e., method α). We will then need the following identification scheme for the operators. Index each operator in the original structure by the number of the components in the original mixture which are lighter than the heavy key for that operator. Thus α_j is an operator denoting a separator with j basic components lighter than the heavy key of the split for that separator. Next we note that operator α_j in the new structure is that operator which represents the same split as in the original structure and thus also has j components lighter than its heavy key. Figs. 27A and 27B illustrate the application of rule 1 with the operators identified in both.

Identifying the operators in this manner, we observe that in all flowsheets, α_j never changes its vertical position relative to all other operators. The difference in flowsheets can be characterized by where the operators are relative to each other in the horizontal direction only. Thus an operator α_j is either before α_k , after α_k ,

or on a path parallel to it. It is always above if $j < k$. The algorithm then for step II follows:

- A. Label all basic operands as "completed".
- B. In the target flowsheet locate an operator (let it be α_k) with two "completed" operands. Clearly at least one exists.
- C. Locate the same operator in the current flowsheet.
- D. If the up operand in the current flowsheet is not a completed operand, apply rule 1 to α_k . Repeat at most $k-1$ times. (In $k-1$ or fewer steps the up operand must become a completed operand as each application of rule 1 reduces by one the number of operators above and to the right of α_k . At most $k-1$ such operators can exist in the current flowsheet.)
- E. If the down operand is not completed apply rule 2 to α_k . Repeat at most $n-k-2$ times (where $n-1$ is the number of operators in the flowsheet).
- F. The operator α_k now has two completed operands (as required in the target flowsheet). Label α_k and its two completed operands a "completed operand" in the target and the flowsheet modified during the steps D and E and repeat from step B until all $n-1$ operators are labeled as part of completed operands.

Note that once an operand, either complex or basic is labeled "completed" in step F above, rules 1 and 2 as applied in steps D and E will never alter it.

Thus the original flowsheet has been converted one operator at a time into the target. The algorithm is finite since all steps are. This completes the proof of the completeness of the evolutionary rules 1, 2 and 3.

VII.2.e. Evolutionary Strategy

In the previous sections a compact way to represent a separation sequence and a set of evolutionary rules were developed. It still remains to establish a search strategy with which, starting from a good initial flowsheet, we can move quickly and effectively in a systematic way to the optimum.

Repeated application of rules 1, 2 and 3 can generate all the possible separation sequences, since these rules possess the property of completeness. If each flowsheet is optimally sized and costed, the optimum flowsheet is easily identified. In principle this exhaustive enumeration strategy will provide the optimum solution to any size separation problem. However as the number of components in the initial mixture and/or the number of the separation methods considered increases, the number of the possible alternatives grows so large that an exhaustive enumeration becomes prohibitive. Therefore certain guides are required to define the direction which will be most promising for yielding the optimum solution, thus reducing significantly the search space. These guides are heuristic in nature. For the synthesis of separation sequences, the most promising direction is that of the neighboring flowsheet which has the lowest cost. Thus, consider the initial flowsheet and its neighboring flowsheets derived by using rules 1 and 2. These flowsheets are optimally sized and costed and the one that exhibits the lowest cost is retained while the rest are rejected. The neighboring flowsheets for this new base flowsheet are developed and evaluated. We always move towards the direction of the cheapest neighbor if of course this neighbor appreciates a lower cost than the current best

flowsheet. This procedure ends when no up or down neighboring flowsheet to the current best flowsheet has a lower cost. Then rule 3 is applied to generate a new class of neighboring flowsheets. We proceed in the same manner until no neighboring flowsheet can be generated with lower cost. Then we have a local optimum.

This approach can be improved by generating a second level of neighboring flowsheets. Thus neighbors of the neighboring flowsheets to a given flowsheet are generated and evaluated. From all the flowsheets the one with the lowest cost is retained. Such a variation requires more computing time but also improves the ability of the strategy to locate the best flowsheet. As mentioned in section VII.1., Pho and Lapidus (1973) used a similar look-ahead strategy with a variable number of look-ahead steps for the synthesis of heat exchange networks. In their approach the solution found depends very much on the number of the look-ahead steps, since whole classes of flowsheets are disregarded once a certain direction is chosen. With the strategy proposed in this work flowsheets are neglected only temporarily but the possibility exists they will be generated later as neighboring flowsheets of other flowsheets.

Another variation of the evolutionary strategy, making use of heuristics and thus potentially requiring less computing time, is the following. The structures for the neighboring flowsheets to the current flowsheet are derived. The most promising is chosen and only this is sized and costed. If the neighbor represents an improvement, it immediately becomes the new basic flowsheet. The selection of the most promising neighbor is heuristic and the success of the method depends on the experience of the designer in the particular area of

the synthesis problem. Heuristics which can help the synthesis of multicomponent separation sequences using this last evolutionary strategy are given by Thompson and King (1972) and King (1971).

VII.3. Examples of Evolutionary Synthesis

In the following two sections the ideas developed previously will be tested on the synthesis of separation systems. The first example is the synthesis of a separation train for solids using two methods of separation, screening and the use of dense liquids in which some of the rocks float and others sink. It is a simple example of large size (1344 flowsheets possible) where the main interest is to demonstrate the use of the evolutionary rules and of the evolutionary strategy as developed in the previous sections. The second example is the synthesis of a liquid multicomponent separation scheme for a real industrial problem. It was taken from Hendry and Hughes (1972) and represents a real process (Buell and Boatright, 1947) as encountered by the practicing engineer. Unlike the first example the separation units are more complicated; their computation time is longer. More complex cost functions were used and thermodynamic routines were needed to calculate the thermodynamic properties of the mixture. While the first example was solved by hand, the second required the use of a computer.

VII.3.a. Example 1: synthesis of a solids' separation system

The problem is to separate a mixture of solids into separate homogeneous piles consisting of the same type of solids. Consider a mixture of 100 tons of solids A,B,C,D,E and F. Each type of solids has a different size and different specific gravity. The

specifications of this mixture of solids and the corresponding sizes and specific gravities for each type are shown in Table 8. We shall consider two separation methods, the use of screens (method α) and the use of dense liquids in which some of the rocks float and others sink (method β). It is clear that a screen can separate the rocks put across it into two piles: those which get through the screen go into one pile and those which are too large go into the other. Using a dense liquid also gives two resulting piles: those which are less dense than the liquid and float and the rest which sink. Figure 33 illustrates a structure, i.e., a system of separators, which could separate the pile of rocks into six piles each containing only one type of rock.

We shall use the following cost functions for the two types of separation methods:

$$C_{\alpha} = \text{weight of the input stream handled ($)}$$

$$C_{\beta} = \text{weight of the input stream / (specific gravity difference of the two rock types next to the point of the split) ($)}$$

The cost of the structure in Figure 33 is \$236 with individual costs as follows:

Separator α_3	Cost: \$ 100
----------------------	--------------

Separator β_1	Cost: \$ 60
---------------------	-------------

Separator α_2	Cost: \$ 15
----------------------	-------------

Separator β_4	Cost: \$ 41
---------------------	-------------

Separator α_5	Cost: \$ 20
----------------------	-------------

TABLE 8
SPECIFICATIONS OF THE SOLIDS IN EXAMPLE 1

<u>Rocks</u>	<u>Diameter (cm)</u>	<u>Specific Gravity (gr/cm³)</u>	<u>Amount(tons)</u>
A	6	2.0	10
B	5	1.5	15
C	4	3.7	5
D	3	1.8	50
E	2	4.1	10
F	1	3.5	10

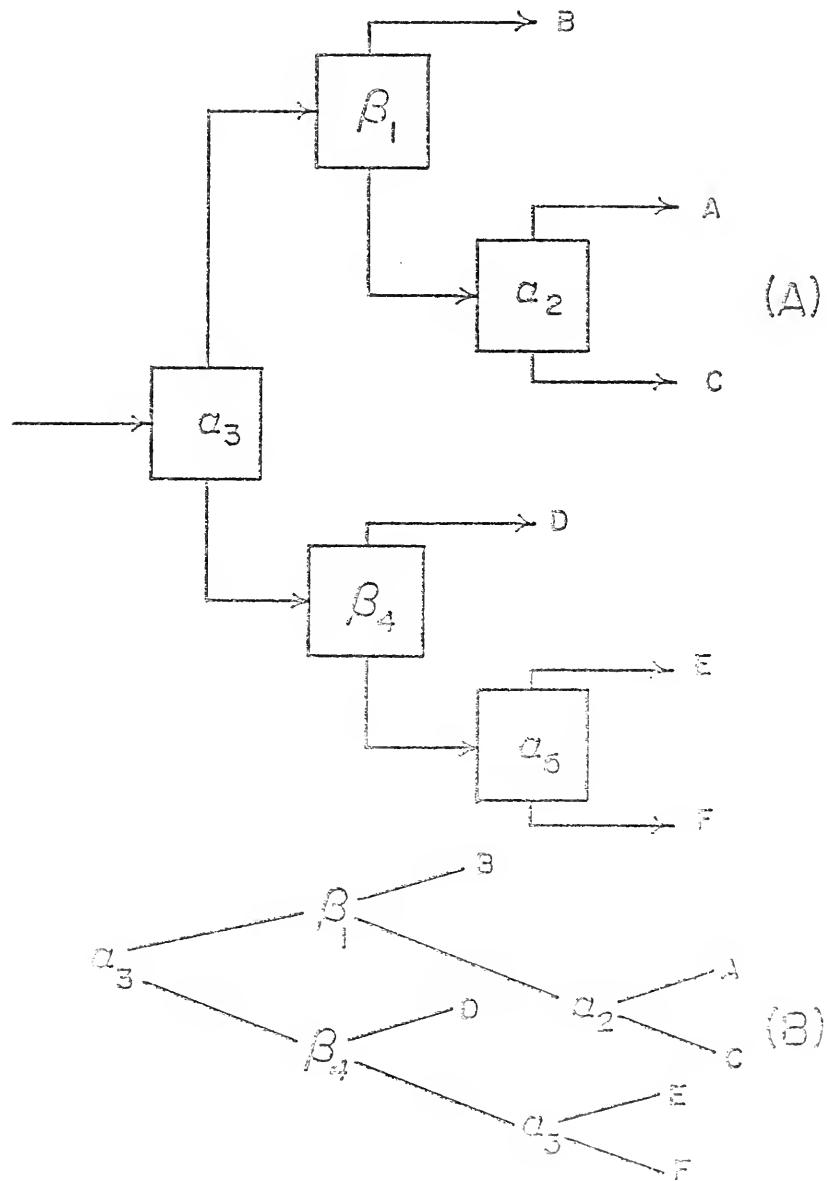


Figure 33. A Schematic Representation of a System for the Separation of Solids and the Corresponding Binary Tree

The evolutionary rules 1, 2 and 3 developed earlier were used for the evolutionary synthesis of the above system.

Applying rules 1 or 2 suggests that we interchange α_3 and β_1 , β_1 and α_2 , α_3 and β_4 , or β_4 and α_5 , thus creating four neighboring structures. Rule 3 would result in changing an α separator into a β separator or the reverse; β_1 becomes α_1 , α_2 becomes β_2 , and so forth. Five additional neighboring structures can be created.

Twelve interchanges were made in the order indicated in Table 9 to locate a local minimum structure of \$200.3. An exhaustive enumeration of all the feasible flowsheets through a branch and bound strategy showed that this is also the global solution to the problem.

VII.3.b. Synthesis of a multicomponent separation scheme

The n-butylene purification system, taken from Hendry and Hughes (1972) and discussed in Chapter VI, will be used to illustrate further the principles of the systematic evolutionary procedure developed here. The initial feed contains Propane (C3), n-Butane (NB), Butene-1 (B1), trans-Butene-2 (B2T), Cis-Butene-2 (B2C) and Pentane (C5). Distillation (α) and extractive distillation (β) will again be the separation methods considered.

To generate the initial flowsheet we will use the strategy employed to establish the basic flowsheet in the branch and bound method in Chapter VI. Thus the sublagrangians of the 6-component separators are evaluated and the one with the least value is chosen as the first separator. In a similar manner we find the second, third and fourth separators for the initial flowsheet. The basis of such an approach is the fact that the separation units, being effectively constrained,

TABLE 9
THE EVOLUTIONARY STEPS TAKEN DURING
THE SYNTHESIS OF THE SOLIDS SEPARATION SYSTEM

<u>Number</u>	<u>Modification</u>	<u>Rule</u>	<u>Cost</u>	<u>Comment</u>
1	$\beta_1 - \alpha_2$	1	241	Reject
2	$\alpha_3 - \beta_1$	2	494	Reject
3	$\alpha_3 - \beta_4$	1	910	Reject
4	$\beta_4 - \alpha_5$	1	271	Reject
5	$\beta_1 - \alpha_1$	3	211	Incorporate
6	$\alpha_1 - \alpha_2$	2	216	Reject
7	$\alpha_3 - \alpha_1$	2	271	Reject
8	$\alpha_2 - \beta_2$	3	200	Incorporate
9	$\alpha_1 - \beta_2$	1	203	Reject
10	$\alpha_3 - \beta_3$	3	498	Reject
11	$\alpha_4 - \beta_4$	3	229	Reject
12	$\alpha_5 - \beta_5$	3	213	Reject

yield values for the dual bounds which are insensitive to the Lagrange multipliers and do not differ very much from the optimal primal values. For the n-butylene purification system the initial flowsheet thus created is the flowsheet (a) in Figure 34. The minimum cost for this flowsheet is found to be \$884,828 / yr.

If we apply rules 1 and 2 on the operators of this flowsheet, the neighboring flowsheets (b), (c), and (d) are generated. These flowsheets and their minimum costs are shown in Figure 34. Flowsheet (c) has the lowest cost of all the neighboring flowsheets, thus we move from (a) to (c) which becomes the new basic flowsheet. The neighbors of (c) are flowsheets (a), (e) and (f). We move to flowsheet (e) which has the lowest cost. None of the neighbors of the flowsheet (e), flowsheets (g) and (h) which are generated by using rules 1 and 2, appreciate a lower cost than the cost for (e). At this point we generate the neighbors of (e) using rule 3, obtaining flowsheets (i) and (j). Similarly, we find out that none of these new neighbors has a lower optimum cost than (e), thus we conclude that flowsheet (e) is a local optimum. In fact this is the globally optimum flowsheet as shown by Hendry and Hughes through an exhaustive enumeration guided by Dynamic Programming. Figure 35 shows the evolution from flowsheet (a) to the optimum flowsheet (e).

Starting with all possible initial flowsheets the same optimum, flowsheet (e), was always located. In Fig. 36 the evolution is shown from flowsheet (k) which is a very bad starting point to the optimum, while the newly generated flowsheets are shown in Figure 37.

For this simple problem the maximum number of flowsheets that had to be optimized by starting with the worst possible first flowsheet

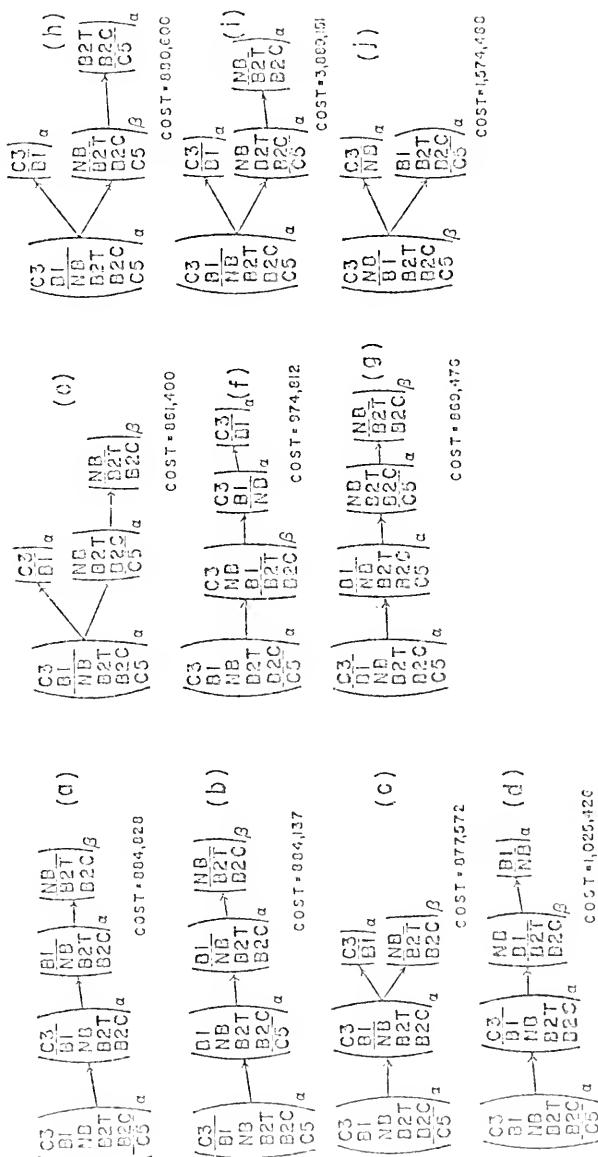


Figure 34. Flowsheets Generated During the Evolutionary Synthesis of the n-Butylene Purification System Starting from Flowsheet (a)

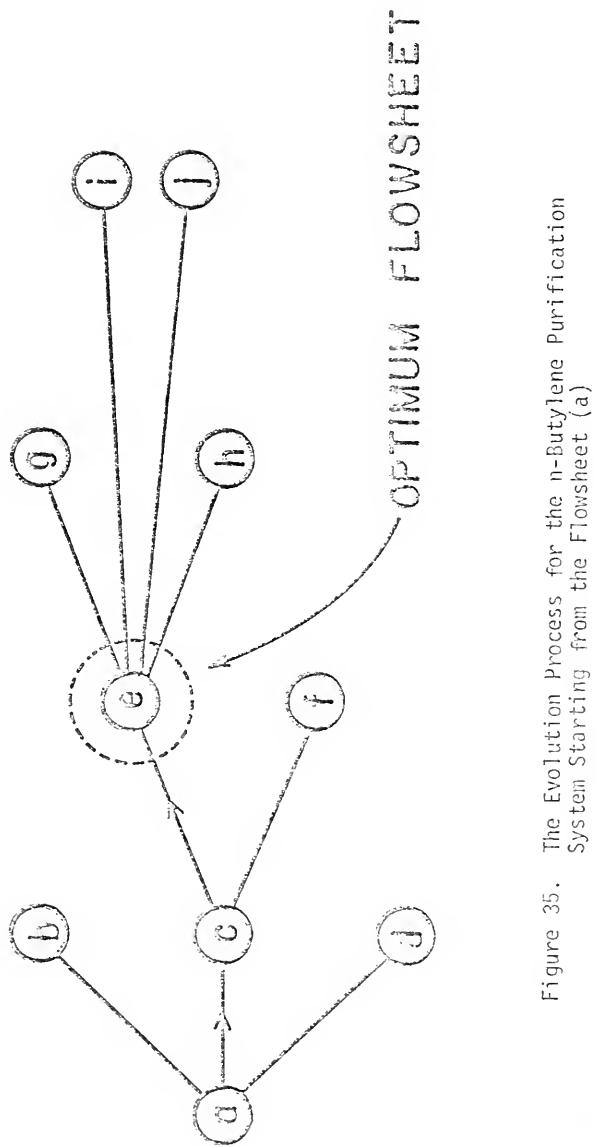


Figure 35. The Evolution Process for the n-Butylene Purification System Starting from the Flowsheet (a)

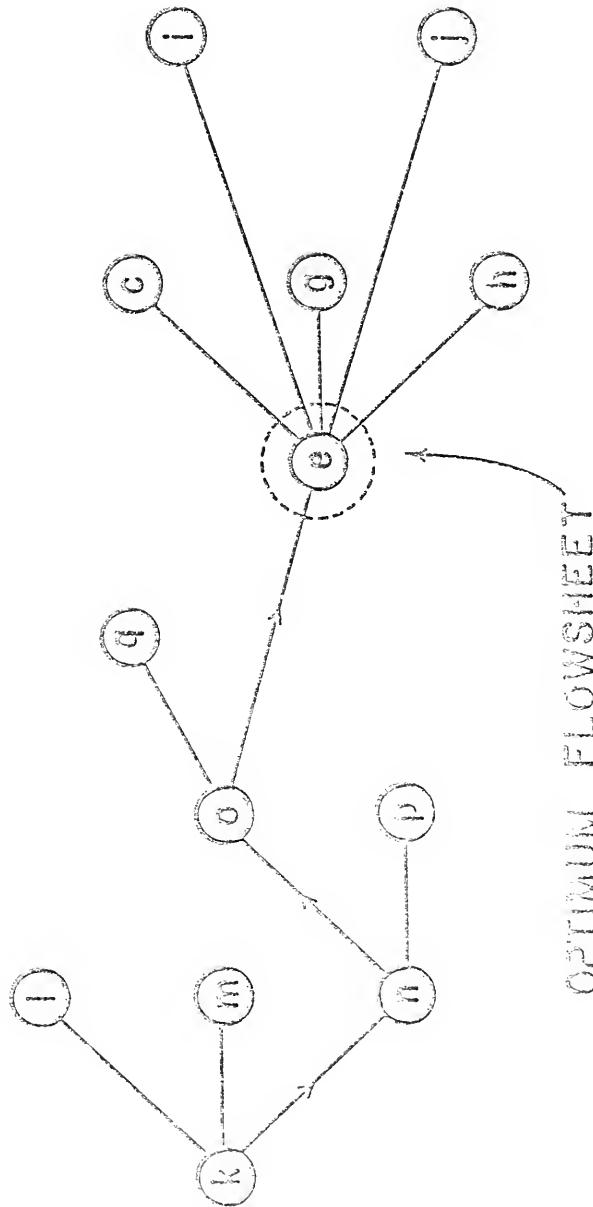


Figure 36. The Evolution Process for the n-Butylene Purification System Starting from the Flowsheet (k)

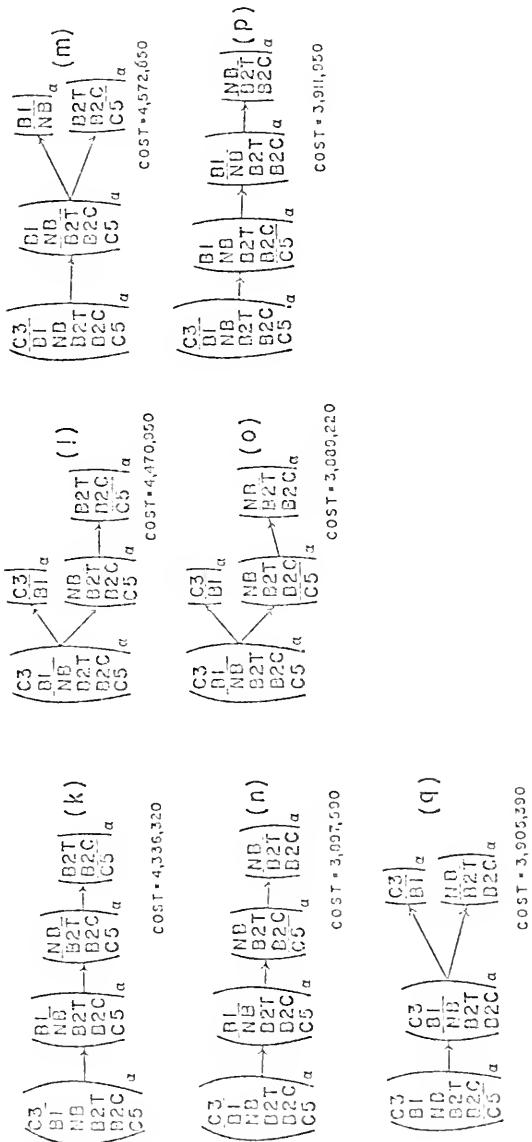


Figure 37. Flowsheets Generated During the Evolutionary Synthesis of the n-Butylene Purification System Starting from Flowsheet (k)

was 19. This number should be compared to 227, the total number of flowsheets which can be generated (Hendry and Hughes, 1972), to give a measure of the success of the proposed method to this particular synthesis problem.

VII.4. Discussion

An approach to organize and systematize the evolutionary synthesis of process flowsheets has been presented. The principal value of the present work lies in the fact that evolutionary thinking of the designer can be structured in a logical manner so that it does not depend solely on his intuition. The systematic development of structural modifications through evolutionary rules and the establishment of favorable or more promising directions are the two major factors of the proposed approach. A first attempt to synthesize optimal multicomponent separation sequences has been successful. Work is currently in progress to develop evolutionary rules for heat exchange networks which also allow for stream splitting and for integrated separation schemes with heat recovery networks.

The experience with the synthesis of separation schemes has been encouraging. Promising directions, once established, move the process quickly to very attractive flowsheets. Information accumulated during the beginning exploratory steps of the evolution can be used very effectively in a heuristic sense to accelerate the process and to limit the search space. In this respect evolutionary synthesis has the advantage of an information feedback procedure to improve the effectiveness of the decision making at a given stage of the synthesis problem.

The reasonability of the evolutionary approach can be demonstrated by comparing the total number of feasible flowsheets for a given separation problem to the number of flowsheets examined. Thus, for an initial mixture of n components using only one method of separation, Table 10 demonstrates the effectiveness of the proposed separation sequence. Table 11 corresponds to a situation where three separation methods are used. In both tables the column "Ratio" denotes how many flowsheets are not examined per flowsheet examined. Thus for $n=6$ and one separation method, we examine on the average at most 22 flowsheets out of the 42 possible. For the calculation of the number of possible flowsheets the following formula was used (Thompson and King, 1972):

$$F = \frac{(2(n-1))!}{n!(n-1)!} \times S^{n-1}$$

where S is the number of the employed separation methods. The assumptions for the estimation of an average search and the formulas giving the number of the examined flowsheets during an average search are given in the Appendix A.

With this approach matched by an efficient strategy for developing good initial flowsheets, it may prove to be a very useful tool in the hands of a designer. It is not intended only as a completely computerized synthesis strategy but also as an approach where the process design engineer, interacting with the computer, organizes and structures his thinking as a means to achieve "optimum" solutions.

TABLE 10
 TABLE INDICATING THE EFFECTIVENESS OF THE
 PROPOSED EVOLUTIONARY STRATEGY IN SYNTHESIZING
 AN OPTIMAL SEPARATION SEQUENCE (ONE SEPARATION METHOD USED)

<u>Number of Components in the Initial Mixture</u>	<u>Number of Feasible Separation Sequences</u>	<u>Number of Flowsheets Examined During an Evolutionary Search</u>			<u>Ratio</u> $F/M_A = 2$
		<u>Smallest</u>	<u>Largest</u>	<u>Average</u>	
3	$F = 2$	$M_S = 1$	$M_L = 1$	$M_A = 1$	1.99
6	42	4	40	22	
9	2,860	7	196	101.5	163.4
13	1,410,864	11	726	368.5	3,828.7
15	2,674,410	13	1,183	598.	4,489.

TABLE II

TABLE INDICATING THE EFFECTIVENESS OF THE PROPOSED
EVOLUTIONARY STRATEGY IN SYNTHESIZING AN OPTIMAL
SEPARATION SEQUENCE (THREE SEPARATION METHODS USED)

<u>Number of Components in the Initial Mixture</u>	<u>Number of Feasible Separation Sequences</u>	<u>Number of Flowsheets Examined During an Evolutionary Search</u>	<u>Ratio</u> $F/M_A =$
n = 3	$F = 2 \times 9$	$M_A = 4$	$F/M_A = 4.5$
6	42×243	220	46.4
9	$2,860 \times 6,561$	1,624	11,554.5
13	$1,410,864 \times 531,441$	8,844	$85 \cdot 10^6$
15	$2,674,410 \times 4,782,969$	16,744	$763 \cdot 10^6$

CHAPTER VIII

CONCLUSIONS AND RECOMMENDATIONS FOR FURTHER RESEARCH

In this dissertation two problems in chemical engineering process design were tackled. The first involved the optimization of nonconvex systems, very commonly encountered in chemical engineering, using modified version of the two-level optimization method and the discrete minimum principle. The second was concerned with the development of efficient strategies for the synthesis of process flowsheets.

Hestenes' method of multipliers was employed successfully to overcome the deficiencies of the two-level optimization procedure and to develop a stronger version of the discrete minimum principle. The shortcomings for both methods are caused by the presence of non-convexities either in the objective functions or the constraints on the units of the flowsheet.

The essence of the proposed method is that the nature of the stationary points of the Lagrangian function can be altered by introducing a penalty term. Thus, maxima or saddle points can be turned to minima points by simply increasing the value of the penalty constant. This secures the existence of a saddle point for the Lagrangian function and therefore guarantees the solution of the initial problem through the use of a Lagrangian based method. The mathematical assumptions which were made to prove the validity of

the proposed method are satisfied by most of the systems of interest. The new procedure has been illustrated by several numerical examples and its usefulness for the chemical engineering process design has been demonstrated on the design of a heat recovery network. Although these illustrations and demonstrations concern rather small examples for which the computations were made by hand, they are not sufficient to establish the general effectiveness of the methods, from a computation-time-required point of view, for large problems. Thus we suggest that a study of the computational characteristics of the new methods be undertaken for large problems. At the present an example of a somewhat larger problem is under way (Stephanopoulos and Westerberg, 1974) concerning the synthesis of an integrated system. The stronger version of the discrete minimum principle is probably faster than the new algorithm for the two-level optimization method because the problem is always feasible and does not require a second level of iteration. This second level of iteration does not require exceedingly lengthy computation times because, for a large enough value of the penalty constant, the sub-lagrangians are likely well behaved and the minimum points should be found quickly. As it has already been mentioned the introduction of the penalty term desensitizes the Lagrangian function with respect to the Lagrange multipliers. This characteristic is of great importance in situations where the primal and the dual values are used as upper and lower bounds of the objective function respectively, as in the case with the branch and bound strategy for the synthesis of optimal multicomponent separation sequences described in Chapter VI. The loss of separability (which is only overcome computationally) will however mean that the

penalty term can only be used to refine complete flowsheet dual bounds. The units will be tied together at the solution.

With respect to the synthesis of process flowsheets two general procedures were developed. The first makes use of the bounding properties of the primal and dual values in a branch and bound scheme to synthesize the optimum sequences of multicomponent separation problems. The second develops a systematic approach to evolutionary synthesis.

The branch and bound strategy was applied to two separation problems with satisfactory results. The number of the optimized separators was considerably reduced compared with previous approaches while the flowsheets generated were nearly optimum so that further screening was not necessary and the decision for the best could be made based on other considerations. Thus for the n-butylene purification system, from 64 possible separators, only 43 separators were optimized to generate 4 nearly optimum flowsheets out of 227 possible. For the olefins, paraffins separation system, from 70 possible separators, only 36 were optimized to generate 5 nearly optimum flowsheets. As mentioned in Chapter VI the success of the branch and bound strategy depends on the good solution of the following three problems: a) the generation of a good basic flowsheet, b) the computation of a satisfactory primal bound and c) the good estimation of the initial Lagrange multipliers. There exists fertile and promising ground for each of the above three problems. Thus it is suggested that new techniques be developed, or existing techniques be tested in their effectiveness to produce satisfactory basic flowsheets for the synthesis of general integrated process systems. The AIDES system

(Siirala, Powers, and Rudd, 1971) or extensions of it may very well constitute a good approach in establishing a basic flowsheet. The computation of a satisfactory primal bound for the basic flowsheet is not a trivial problem. Sometimes repeated iterations are needed to establish a good, tight primal bound. One approach which suggests itself, as McGalliard (1971) noted, is to refine the information reversal procedure by applying a reversal sensitivity analysis at the point of occurrence of system infeasibility in the primal calculation. Clearly this would be a local procedure and could incur a sizable computational load, but a modest improvement in the effectiveness of the resulting primal bound could justify its use. The problem of good estimation of the Lagrange multipliers is not critical for the synthesis of separation sequences since the decision variables are tightly constrained. But for more general systems which do not have the above property, the estimation of good initial Lagrange multipliers becomes very important. One alternative is to introduce in the problem a penalty term which will desensitize the dependence of the dual value on the Lagrange multipliers. This would slow down the procedure. It is suggested that new alternative methods to estimate good initial Lagrange multipliers be developed.

The work on evolutionary synthesis has paved the ground for more extended research in this area. The basic principles and notions have been underlined and preliminary work has shown the usefulness of the approach. Further work is needed to establish evolutionary rules for other systems such as heat exchangers, reactor networks, separation sequences with heat recovery networks etc. The simple examples on which evolutionary synthesis was illustrated demonstrate that the

evolutionary approach may constitute a powerful tool for this purpose. Further work is needed to verify this preliminary result in more complicated integrated systems.

APPENDICES

APPENDIX A

In section VII.4. the reasonability and the efficiency of the evolutionary procedure was demonstrated by comparing the total number of feasible flowsheets for a given separation problem to the number of flowsheets examined during an average search. Thus, for an initial mixture of N components using S methods of separation, the number of the feasible flowsheets is given by the following formula:

$$F = \frac{(2(N-1))!}{N! (N-1)!} S^{N-1}$$

In order to estimate the number of flowsheets examined during an average evolutionary search, some assumptions must be made.

The first and the basic assumption is that we will generate and evaluate all the possible neighbors using rules 1, 2 and 3 at each step of the evolution. For a mixture of N components using only one method of separation, the smallest number of the neighboring flowsheets that can be generated during the evolutionary procedure is the number of pairwise interchanges to prove the initial flowsheet is optimal:

$$n_S = (N-2)$$

The largest number of examined flowsheets is

$$n_L = (N-2) (N-1) (N-2) / 2$$

because as assumed, at each evolutionary step, all possible neighbors are generated by $(N-2)$ pairwise interchanges using rules 1 and 2.

Also for each of the above nieghbors there will be at most

$$(N-2) + (N-3) + \dots + 1 + 0 = (N-2)(N-1)/2$$

(see step D of the algorithm in section VII.2.d.) consecutive structural changes to produce any of the flowsheets. Therefore the actual number of the flowsheets examined is between these two bounds and on the average we assume the arithmetic mean

$$M_A = (N-2) [(N-2)(N-1) + 2] / 4$$

When more than one separation method is considered, the number of flowsheets examined at each step in an average evolutionary search increases by the factor

$$(N-1)(S-1)$$

since the application of rule 3 at each stage generates $(N-1)(S-1)$ new flowsheets.

In Table 10 the smallest (M_S), the largest (M_L) and the average (M_A) numbers of flowsheets examined are shown for mixtures of N components (with different values of N) and employing only one separation method, while in the Table 11 M_A is shown for cases where three separation methods have been considered. Values of the F/M_A ratio in the above tables show the potential effectiveness of the evolutionary procedure.

These estimates of the average numbers of the flowsheets examined during an evolutionary synthesis of a multicomponent separation sequence are rather conservative estimates. There are two reasons. First, one is likely to start the evolution with a flowsheet which is not far from the optimum. This will eliminate a large number of flowsheets which need not be considered. Second, it is also likely that one will not use the strategy to generate and evaluate all neighboring flowsheets. The evolutionary strategy may be based on the examination of the neighbors which are generated by a subset of the evolutionary rules or move to the first improved flowsheet found and only when we find a local minimum flowsheet we will evaluate most or all of the neighbors. The strategy to use only a subset of the rules was applied in the second example synthesized by evolution in Chapter VII and to move to the first improved flowsheet found in the first example.

BIBLIOGRAPHY

Avery, C.J., and A.S. Foss, "A Shortcoming of the Multilevel Optimization Technique," *AIChEJ.*, 17 (4), p. 998 (1971).

Aris, R., *Discrete Dynamic Programming*, Blaisdell, N.Y. (1964).

Bellman, R., and W. Karush, "On a New Functional Transform in Analysis: The Maximum Transform," *Bulletin of American Mathematical Society*, 67 (5), p. 501 (1961).

Bellmore, M., H.J. Greenberg, and J.J. Jarvis, "Generalized Penalty Function Concepts in Mathematical Optimization," *Oper. Res.*, 18 (2), p. 229 (1970).

Brosilow, C., and L. Lasdon, "A Two-Level Optimization Technique for Recycle Processes," *AIChE-I.Chem.E. Symposium Series*, No. 4 (1965).

Buell, C.K., and R.G. Boatright, "Furfural Extractive Distillation for Separation and Purification of C_4 Hydrocarbons," *Ind. Eng. Chem.*, 39 (6), p. 695 (1947).

Denn, M.M., *Optimization by Variational Methods*, McGraw-Hill, New York, N.Y. (1969).

Everett, H. III, "Generalized Lagrange Multiplier Method for Solving Problems of Optimum Allocation of Recourses," *Oper. Res.*, 4 (3), p. 399 (1963).

Falk, J.E., and R.M. Soland, "An Algorithm for Separable Nonconvex Programming Problems," *Manag. Science*, 15 (9), p. 560 (1969).

Fan, L.T., and C.S. Wang, *The Discrete Maximum Principle*, Wiley, New York, N.Y. (1964).

Glover, F., "A Multiphase-Dual Algorithm for Zero-One Integer Programming Problems," *Oper. Res.*, 16 (4), p. 883 (1968).

Gould, F.J., "Extensions of Lagrange Multipliers in Nonlinear Programming," *S.I.A.M. Journal, Applied Mathematics*, 17 (6), p. 1280 (1969).

Greenberg, H.J., "Lagrangian Duality Gaps: Their Source and Resolution," Technical Report CP-69005, Southern Methodist University, April (1969).⁴

Greenberg, H.J., "Bounding Nonconvex Programs by Conjugates," *Oper. Res.*, 21 (1), p. 346 (1973).

Greenberg, H.J., and W.P. Pierskalla, "Surrogate Mathematical Programs," *Oper. Res.*, 18 (5), p. 483 (1970).

Halkin, J., "A Maximum Principle of the Pontryagin Type for Systems Described by Nonlinear Difference Equations," *S.I.A.M. Journal, Control*, 4 (1), p. 90 (1966).

Hendry, J.E., and R.R. Hughes, "Generating Separation Process Flowsheets," *Chem. Eng. Progr.*, 68 (6), p. 69 (1972).

Hendry, J.E., D.F. Rudd and J.D. Seader, "Synthesis in the Design of Chemical Processes," *AIChEJ.*, 19 (1), p. 1 (1973).

Hestenes, M.R., "Multiplier and Gradient Methods," *J. of Optimiz. Theory and Applications*, 4 (5), p. 303 (1969).

Holtzman, J.M., and H. Halkin, "Directional Convexity and the Maximum Principle for Discrete Systems," *S.I.A.M. Journal, Control*, 4 (2), p. 263 (1966).

Horn, F., "Über das Problem der Optimalen Rührkesselkaskade für Chemische Reaktionen," *Chem. Eng. Sci.*, 15 (2), p. 176 (1961).

Horn, F., and R. Jackson, "Discrete Maximum Principle," *Ind. and Eng. Chem., Fundamentals*, 4 (1), p. 110 (1965a).

Horn, F., and R. Jackson, On Discrete Analogues of Pontryagin's Maximum Principle," *Intern. J. on Control*, 1 (3), p. 389 (1955b).

Ichikawa, A., and L.T. Fan, "Optimal Synthesis of Process Systems. Necessary Conditions for Optimal System and Its Use in Synthesis of Systems," *Chem. Eng. Sci.*, 28 (2), p. 357 (1973).

Jackson, R., "Some Algebraic Properties of Optimization Problems in Complex Chemical Plants," *Chem. Eng. Sci.*, 19 (1), p. 19 (1964).

Katz, S., "Best Operating Points for Staged Systems," *Ind. and Eng. Chem., Fundamentals*, 1 (2), p. 226 (1962).

King, C.J., *Separation Processes*, McGraw-Hill Book Co., New York, N.Y. (1971).

King, C.J., D.W. Gantz and F.J. Barnes, "Systematic Evolutionary Process Synthesis," *Ind. and Eng. Chem., Process Design and Development*, 11 (2), p. 271 (1972).

Knuth, D.E., *The Art of Computer Programming*. Vol. 1, Addison-Wesley Publishing Company, Reading, Mass. (1968).

Lasdon, L.S., "A Multi-Level Technique for Optimization," Ph.D. Thesis, Case Institute of Technology (1964); Systems Res. Center Report SRC 50-C-64-19.

Lasdon, L.S., *Optimization Theory for Large Systems*, MacMillan Publishing Co., New York, N.Y. (1970).

Lee, K.F., A.H. Masso, and D.F. Rudd, "Branch and Bound Synthesis of Integrated Process Designs," Ind. and Engin. Chem., Fundamentals, 9 (1), p. 48 (1970).

Loane, E.P., "An Algorithm to Solve Finite Separable Constraint Optimization Problems," Oper. Res., 19 (6), p. 1477 (1971).

Masso, A.H., and D.F. Rudd, "The Synthesis of System Designs: II. Heuristic Structuring," AIChEJ., 15 (1), p. 10 (1969).

McGalliard, R.L., "Structural Sensitivity Analysis in Design Synthesis," Ph.D. Thesis, University of Florida (1971).

McGalliard, R.L., and A.W. Westerberg, "Structural Sensitivity Analysis in Design Synthesis," Chem. Eng. J., 4 (4), p. 127 (1972).

Miele, A., P.E. Moseley, A.V. Levy and G.M. Coggins, "On the Method of Multipliers for Mathematical Programming Problems," J. of Optimiz. Theory and Applications, 10 (1), p. 1 (1972).

Osakada, K., and L.T. Fan, "Synthesis of an Optimal Large-Scale Interconnected System by Structural Parameter Method Coupled with Multilevel Technique," Can. J. of Chem. Eng., 51 (1), p. 94 (1973).

Pho, T.K., and L. Lapidus, "Topics in Computer-Aided Design: Part II. Synthesis of Optimal Heat Exchanger Networks by Tree Searching Algorithms," AIChEJ., 19 (6), p. 1182 (1973).

Pontryagin, L.S., V.A. Boltyanskii, R.V. Gamkrelidze, and E.F. Mischenko, *The Mathematical Theory of Optimal Processes*, John Wiley and Sons, New York, N.Y. (1962).

Powers, G.J., and R.L. Jones, "Reaction Path Synthesis Strategies," AIChEJ., 19 (6), p. 1204 (1973).

Rockafellar, R.T., "Non-Linear Programming," American Mathematical Society Summer Seminar on the Mathematics of the Decision Sciences, Stanford University, July-August (1967).

Rozonoer, L.I., "The Maximum Principle of L.S. Pontryagin in Optimal System Theory: Part III.," Automation Remote Control, 20 (12), p. 1517 (1959).

Rudd, D.F., "The Synthesis of System Designs: I. Elementary Decomposition Theory," AIChEJ., 14 (2), p. 343 (1968).

Schock, A.V., and R. Luus, "Relationship of the Two-Level Optimization Procedure to the Discrete Maximum Principle," AIChEJ., 18 (3), p. 659 (1972).

Sirola, J.J., and D.F. Rudd, "Computer-Aided Synthesis of Chemical Process Designs," Ind. and Eng. Chem., Fundamentals, 10 (3), p. 353 (1971).

Sirola, J.J., G.J. Powers, and D.F. Rudd, "Synthesis of Systems Designs: III. Toward a Process Concept Generator," AIChEJ., 17 (3), p. 677 (1971).

Soland, R.M., "An Algorithm for Separable Nonconvex Programming Problems: II. Nonconvex Constraints," Manag. Science, 17 (11), p. 750 (1971).

Stephanopoulos, G., and A.W. Westerberg, "A Multilevel Technique for the Synthesis of Integrated Systems in the Presence of Nonconvexities," in preparation (1974).

Thompson, R.W., and C.J. King, "Synthesis of Separation Schemes," Technical Report, AEC Contract No. W-7405-eng-48, Lawrence Berkeley Laboratory, July (1972).

Umeda, T., A. Shindo, and E. Tazaki, "Optimal Design of Chemical Process by Feasible Decomposition Method," Ind. and Eng. Chem., Process Des. and Develop., 11 (1), p. 1 (1972).

Umeda, T., and A. Ichikawa, "Synthesis of Optimal Processing Systems by a Method of Decomposition," Paper presented in the 71st Nat. Mtg. AIChE, Dallas Texas, February (1972).

Umeda, T., A. Hirai, and A. Ichikawa, "Synthesis of Optimal Processing Systems by an Integrated Approach," Chem. Eng. Sci., 27 (5), p. 795 (1972).

Westbrook, G.T., "Use This Method to Size Each Stage for Best Operation," Hydrocarbon Processing Petroleum Refiner, 40 (1), p. 201 (1961).

Westerberg, A.W., *Decomposition of Large-Scale Problems*, D.M. Himmelblau, editor, pp. 379-397, North Holland/American Elsevier, Amsterdam (1973).

BIOGRAPHICAL SKETCH

George Stephanopoulos, the oldest son of Nicholas and Elisabeth, was born June 1, 1947, in Kalamata, Messinia, Greece. Upon his graduation from the Paralia Gymnasium in June, 1965, he began his university studies at the National Technical University of Athens, Athens, Greece. In June, 1970, following the receipt of his Diploma in Chemical Engineering, he entered the Graduate School of McMaster University, Hamilton, Ontario, Canada, to work for a Master of Engineering degree in Chemical Engineering, which was granted to him in September, 1971.

In September, 1971 he enrolled in the Graduate School of the University of Florida where he was initially awarded a teaching assistantship in the Department of Chemical Engineering. In June, 1972, he was awarded a research assistantship under Dr. A.W. Westerberg and in September, 1973, a Graduate School Research Assistantship for the completion of his studies at the University of Florida.

He was the recipient of the following awards: 1. First Award of the Greek Mathematical Society (1965). 2. Study Scholarships from the National Scholarships Establishment (1965-70). 3. Progress Scholarships from the National Technical University (1967-70). 4. Texaco Oil Co. research fellowship (1969). 5. First Award for Graduating Students of the National Bank of Greece (1970). 6. "THOMAIDION" Award from National Technical University (1970).

7. "CHRYSOVERGION" Award from National Technical University (1970).

Mr. Stephanopoulos has accepted a faculty position as an assistant professor in Chemical Engineering at the University of Minnesota.

I certify that I have read this study and that in my opinion it conforms to acceptable standards of scholarly presentation and is fully adequate, in scope and quality, as a dissertation for the degree of Doctor of Philosophy.



Arthur W. Westerberg, Chairman
Associate Professor of Chemical Engineering

I certify that I have read this study and that in my opinion it conforms to acceptable standards of scholarly presentation and is fully adequate, in scope and quality, as a dissertation for the degree of Doctor of Philosophy.



Michael E. Thomas
Professor of Industrial and Systems
Engineering

I certify that I have read this study and that in my opinion it conforms to acceptable standards of scholarly presentation and is fully adequate, in scope and quality, as a dissertation for the degree of Doctor of Philosophy.



Thomas E. Bullock
Associate Professor of Electrical Engineering

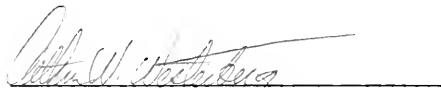
This dissertation was submitted to the Graduate Faculty of the College of Engineering and to the Graduate Council, and was accepted as partial fulfillment of the requirements for the degree of Doctor of Philosophy.

June, 1974

Dean, College of Engineering

Dean, Graduate School

I certify that I have read this study and that in my opinion it conforms to acceptable standards of scholarly presentation and is fully adequate, in scope and quality, as a dissertation for the degree of Doctor of Philosophy.



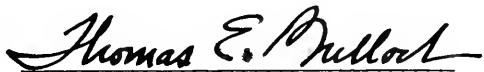
Arthur W. Westerberg, Chairman
Associate Professor of Chemical Engineering

I certify that I have read this study and that in my opinion it conforms to acceptable standards of scholarly presentation and is fully adequate, in scope and quality, as a dissertation for the degree of Doctor of Philosophy.



Michael E. Thomas
Professor of Industrial and Systems
Engineering

I certify that I have read this study and that in my opinion it conforms to acceptable standards of scholarly presentation and is fully adequate, in scope and quality, as a dissertation for the degree of Doctor of Philosophy.



Thomas E. Bullock
Associate Professor of Electrical Engineering

This dissertation was submitted to the Graduate Faculty of the College of Engineering and to the Graduate Council, and was accepted as partial fulfillment of the requirements for the degree of Doctor of Philosophy.

June, 1974



Dean, College of Engineering

Dean, Graduate School